

DLR-IB-RM-OP-2017-295

**Gestaltung eines Augmented
Reality Interaktionskonzepts für die
intuitive Roboterprogrammierung**

Masterarbeit

Annika Wollschläger



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt**

MASTERARBEIT

GESTALTUNG EINES AUGMENTED REALITY INTERAKTIONSKONZEPTS FÜR DIE INTUITIVE ROBOTERPROGRAMMIERUNG

Freigabe:

Der Bearbeiter:

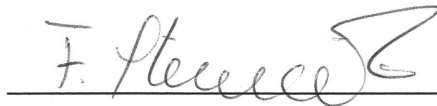
Unterschriften

Annika Wollschläger




Betreuer:

Franz Steinmetz



Der Institutsdirektor

Prof. Alin Albu-Schäffer



Dieser Bericht enthält 99 Seiten, 27 Abbildungen und 4 Tabellen

Technische Universität München

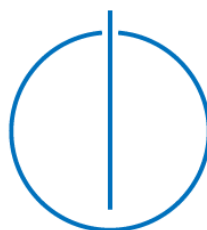


Fakultät für Informatik

Masterarbeit in Informatik: Robotics, Cognition and
Intelligence

Gestaltung eines Augmented Reality Interaktionskonzepts für die intuitive Roboterprogrammierung

Annika Wollschläger



Technische Universität München



Fakultät für Informatik

Masterarbeit in Informatik: Robotics, Cognition and
Intelligence

Gestaltung eines Augmented Reality Interaktionskonzepts für die intuitive Roboterprogrammierung

Design of an Augmented Reality Interaction Concept for Intuitive Robot Programming

Annika Wollschläger

Aufgabensteller **Prof. Dr. phil. Klaus Bengler**

Lehrstuhl für Ergonomie

Technische Universität München

Betreuer **Franz Steinmetz (DLR)**

Jakob Reinhardt (TUM)

Jonas Schmidtler (TUM)

15. Oktober 2017

Annika Wollschläger

Gestaltung eines Augmented Reality Interaktionskonzepts für die intuitive Roboterprogrammierung

Masterarbeit in Informatik: Robotics, Cognition and Intelligence, 15. Oktober 2017

Aufgabensteller: Prof. Dr. phil. Klaus Bengler

Betreuer: Franz Steinmetz (DLR) Jakob Reinhardt (TUM) und Jonas Schmidtler (TUM)

Technische Universität München

Fakultät für Informatik

Arcisstraße 21

80333 and München

Erklärung

Ich versichere, dass ich diese Masterarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, 15. Oktober 2017

Annika Wollschläger

Zusammenfassung

Intuitiv bedienbare Programmierkonzepte versetzen nicht speziell geschultes Personal in die Lage, Programmieraufgaben zu übernehmen. Sie sind daher insbesondere für kleine und mittlere Unternehmen (KMU) von Interesse, um den Programmieraufwand für die Automation zu verringern.

Die vorliegende Arbeit basiert auf einem integrierten Framework zur aufgabenorientierten Programmierung und Ausführung, das die Programmierung von Robotern mittels einer grafischen Oberfläche ermöglicht, indem einfache Programmsequenzen definiert und parametrisiert werden. Als eine Alternative zur Eingabe mittels eines Touchdisplays wird ein neuartiges Interaktionskonzept basierend auf Augmented Reality (AR) und haptischer Eingabe entwickelt. Eingaben können durch das Interagieren mit dem Endeffektor des zu programmierenden Roboters getätigt werden. Einblendungen kontextabhängiger Informationen über eine AR-Datenbrille direkt am Endeffektor des Roboters leiten den Nutzer durch den Programmiervorgang.

Die Gebrauchstauglichkeit des Interaktionskonzepts wurde anhand subjektiver und objektiver Daten im Vergleich zur Programmierung über die grafische Oberfläche auf einem Touchdisplay evaluiert. Die Ergebnisse lassen darauf schließen, dass das AR-Interaktionskonzept in Situationen, in denen der Nutzer im direkten Kontakt mit dem Roboter ist, eine nützliche Ergänzung zum Touchdisplay darstellt.

Abstract

Intuitive programming concepts allow non-experts to undertake programming tasks. They are therefore of interest for small and medium sized companies to reduce the programming effort required for automation. This work builds upon an integrated framework for task-level programming and task execution allowing a non-expert to program robots via a graphical user interface by defining and parameterizing simple program sequences. As an alternative input method to touch, a novel interaction concept based on augmented reality (AR) and haptic inputs is developed. The robot to be programmed serves as the input, i. e. one can push against its end-effector (e. g., from the left or right) to trigger certain actions. The available haptic input options, which are context specific, are superimposed directly at the end-effector using AR-glasses. We evaluate the usability of this interaction concept in comparison to a tablet computer control in a user study by means of subjective and objective data. The results indicate that the AR interaction concept can be a useful extension to touch interfaces in situations where interaction with the robot is required in either case.

Inhaltsverzeichnis

| | | |
|----------|---|----------|
| 1 | Einleitung | 1 |
| 1.1 | Motivation und Problemstellung | 1 |
| 1.2 | Struktur | 3 |
| 2 | Theorieteil | 5 |
| 2.1 | Ergonomische Aspekte | 5 |
| 2.1.1 | Gebrauchstauglichkeit | 5 |
| 2.1.2 | Bewertungsmethoden | 6 |
| 2.2 | Mensch-Roboter Interaktion | 6 |
| 2.2.1 | Klassifikation von Mensch-Roboter Interaktionen . . . | 7 |
| 2.2.2 | Mensch-Roboter Interaktion für die Roboterprogram- mierung | 9 |
| 2.2.3 | Mensch-Roboter Schnittstellen | 10 |
| 2.2.4 | Gebrauchstauglichkeit von Mensch-Roboter Systemen . | 13 |
| 2.3 | Augmented Reality | 14 |
| 2.3.1 | Stereosehen | 16 |
| 2.3.2 | Anzeigegeräte | 17 |
| 2.3.3 | Tracking | 19 |
| 2.3.4 | Registrierung | 21 |
| 2.3.5 | Rendering | 21 |
| 2.3.6 | Anwendungen in der Mensch-Roboter Interaktion . . . | 23 |
| 2.3.7 | Probleme bei Verwendung von Optical-See-Through- Geräten | 23 |
| 2.3.8 | Gestaltungsrichtlinien | 26 |
| 2.4 | Roboterkinematik | 27 |

| | | |
|----------|---|-----------|
| 3 | Methodenteil | 29 |
| 3.1 | Technische Rahmenbedingungen | 29 |
| 3.1.1 | RAZER | 29 |
| 3.1.2 | Datenbrille | 31 |
| 3.1.3 | Roboter | 32 |
| 3.2 | Konzeption der Mensch-Roboter-Schnittstelle | 34 |
| 3.2.1 | Nutzungskontext | 34 |
| 3.2.2 | Systemanforderungen und -design | 35 |
| 3.2.3 | Komponentenanforderungen und -design | 39 |
| 3.3 | Versuchsdesign | 54 |
| 4 | Ergebnisteil | 59 |
| 4.1 | Subjektive Daten | 59 |
| 4.2 | Objektive Daten | 60 |
| 4.2.1 | Bearbeitungszeiten | 60 |
| 4.2.2 | Bedienfehler | 61 |
| 5 | Diskussion und Zusammenfassung | 63 |
| 5.1 | Diskussion der Ergebnisse | 63 |
| 5.1.1 | Versuchsdesign | 63 |
| 5.1.2 | Objektive Daten | 63 |
| 5.1.3 | Subjektive Daten | 65 |
| 5.2 | Ausblick | 68 |
| 5.3 | Zusammenfassung | 70 |
| A | Anhang | 72 |
| A.1 | Ergebnisse | 72 |
| A.2 | Fragebögen | 74 |
| | Literatur | 81 |

Einleitung

1.1 Motivation und Problemstellung

Industrieroboter ermöglichen Steigerungen der Produktivität und Senkung der Stückzahlkosten. Der Einsatz solcher Automatisierungslösungen ist jedoch für kleine und mittlere Unternehmen (KMU) bisher uninteressant, da herkömmliche Systeme hohe Programmierkosten verursachen und kein flexibler Einsatz möglich ist. Auch in großen Unternehmen ist der Bedarf an flexibel einsetzbaren Automatisierungslösungen mit einer hohen Aufgabenvielfalt durch die Just-In-Time-Produktion, der zunehmenden Zahl von Varianten und der verkürzten Produktlebenszyklen gegeben. Die Ablösung der herkömmlichen Expertensysteme durch intuitiv bedienbare Programmierkonzepte kann zur Senkung der Kosten führen, da auch nicht speziell geschultes Personal in die Lage versetzt wird, Programmieraufgaben zu übernehmen. [50]

Am Deutschen Zentrum für Luft- und Raumfahrt (DLR) wurde ein Framework zur intuitiven Roboterprogrammierung im Fertigungskontext entwickelt [54]. Die Programmierung findet dabei über eine grafische Oberfläche auf einem Tabletcomputer und mit Hilfe von Programmieren durch Vormachen (PdV) statt. Besonderheit der Software ist der Grad der Aufgabenorientierung. Der Nutzer spezifiziert beispielsweise beim Bohren lediglich das Material, die Materialdicke, den Durchmesser und das gewünschte Bohrmuster. Die benötigten Bewegungen des Roboters werden aus diesen Informationen anhand von vorprogrammierten Softwaremodulen automatisiert bestimmt.

Ein Problem der Interaktion mit RAZER ist die Abhängigkeit von der grafischen Oberfläche während des gesamten Programmiervorgangs. Dies führt dazu, dass der Nutzer beim Vormachen der Roboterposen mehrfach zwischen der Bedienung des Tabletcomputers, auf dem sich die grafische Oberfläche befindet, und der Bewegung des Roboterendeffektors wechseln muss. Dazu wird der Tabletcomputer abgelegt, die Aktion am Roboter ausgeführt und anschließend der Tabletcomputer wieder aufgenommen. Der Nutzer muss die genaue Anweisung im Kopf behalten. Während der Tabletcomputer abgelegt ist, ist der Nutzer¹ nicht in der Lage auf etwaige Fehlermeldungen zu reagieren. Dies verletzt Punkt 1 der Kriterien zur Evaluation von Mensch-Roboter Systemen nach Scholtz [48], wie sie in Abschnitt 2.2.4 vorgestellt werden. Für den Nutzer ist nicht erkennbar, wann sein Eingreifen erforderlich ist. Zudem können die Interaktionen nicht effizient bewältigt werden, da durch das Ablegen und Aufnehmen des Tabletcomputers Zeit verloren geht.

Im Rahmen der Masterarbeit wird der Einsatz einer Augmented Reality (AR)-Datenbrille zur Vereinfachung der Roboter-Mensch Interaktion erprobt. Durch die Anzeige kontextabhängiger Informationen wird der Nutzer durch Programmierpassagen geleitet, bei denen er im direkten Kontakt mit dem Roboter ist und den Tabletcomputer nicht zur Hand hat. Der Roboterarm dient als Eingabegerät, indem er vom Nutzer kurz in eine bestimmte Richtung gedrückt wird. Die verfügbaren kontextabhängigen Aktionen werden über die AR-Datenbrille direkt am Roboter eingeblendet.

Es wird erwartet, dass der Nutzer in Situationen, in denen er im direkten Kontakt mit dem Roboter steht, vom AR-Interaktionskonzept profitieren kann und dieses somit eine sinnvolle Ergänzung der Graphical User Interface (GUI) auf dem Tabletcomputer darstellt. Dazu findet eine Evaluierung der

¹Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Die Ausführungen beziehen sich gleichermaßen auf beiderlei Geschlecht.

Gebrauchstauglichkeit des AR-Interaktionskonzepts anhand objektiver und subjektiver Daten statt.

1.2 Struktur

In Abschnitt 2 werden die theoretischen Grundlagen erläutert. Abschnitt 2.1 befasst sich mit ergonomischen Aspekten unter dem Blickwinkel der Gebrauchstauglichkeit und ihrer Bewertungsmethoden. Abschnitt 2.2 beschreibt die Mensch-Roboter Interaktion und ihre Klassifikation, die Anwendung in der Roboterprogrammierung, Mensch-Roboter Schnittstellen und die Gebrauchstauglichkeit von Mensch-Roboter Systemen. Abschnitt 2.3 behandelt die theoretischen Grundlagen der AR mit den Aspekten Stereosehen, Anzeigegeräte, Tracking, Registrierung, Rendering, Anwendungen in der Mensch-Roboter Interaktion und Probleme bei Verwendung von AR mit Optical-See-Through (OST)-Geräten. Den Abschluss des Theorieteils bildet Abschnitt 2.4 über Roboterkinematik.

Im Methodenteil in Abschnitt 3 werden in Abschnitt 3.1 die technischen Rahmenbedingungen erläutert, dazu gehören neben RAZER, dem Framework zur intuitiven Roboterprogrammierung, die technischen Details zur AR-Datenbrille und zum verwendeten Roboter. In Abschnitt 3.2 folgt die Konzeption der in dieser Arbeit entwickelten Mensch-Roboter-Schnittstelle mit Nutzungskontext, Anforderungen und Design des Systems sowie der Komponenten. Abschließend wird in Abschnitt 3.3 das Versuchsdesign beschrieben.

Die Ergebnisse des Versuchs werden anschließend in Abschnitt 4 durch die Darstellung subjektiver Daten in Abschnitt 4.1 und der Bearbeitungszeiten sowie Bedienfehler in Abschnitt 4.2 aufgeführt.

Den Abschluss der Arbeit bilden die Diskussion und Zusammenfassung in Abschnitt 5, die die Deutung der Ergebnisse in Abschnitt 5.1 und einen Ausblick in Abschnitt 5.2 beinhalten.

Theorieteil

2.1 Ergonomische Aspekte

2.1.1 Gebrauchstauglichkeit

Zur Bewertung eines Interaktionskonzeptes wird oft die sogenannte Gebrauchstauglichkeit (engl. *Usability*) herangezogen. Diese ist definiert als „das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“ [12]. Effektivität umfasst die Genauigkeit und Vollständigkeit, mit der das Ziel erreicht wird. Das Verhältnis des eingesetzten Aufwands zur Effektivität wird als Effizienz bezeichnet. Zufriedenheit beschreibt die Freiheit von Beeinträchtigungen sowie die positive Einstellung gegenüber der Nutzung des Systems. Der Nutzungskontext umfasst Benutzer, Arbeitsaufgaben, Arbeitsmittel sowie die physische und soziale Umgebung, in der das Produkt genutzt wird [12].

Um Gebrauchstauglichkeit zu erzielen, empfiehlt DIN EN ISO 9241-110:2008-09 [11] folgende Grundsätze der Dialoggestaltung:

- Aufgabenangemessenheit
- Selbstbeschreibungsfähigkeit
- Erwartungskonformität
- Lernförderlichkeit
- Steuerbarkeit
- Fehlertoleranz
- Individualisierbarkeit

2.1.2 Bewertungsmethoden

Um die Gebrauchstauglichkeit eines Systems zu erfassen, sind Maße für Effektivität, Effizienz und Zufriedenstellung vonnöten. Maße der Effektivität sind etwa der Grad der Zielerreichung oder die durchschnittliche Genauigkeit der abgeschlossenen Aufgabe. Maße der Effizienz sind beispielsweise die Zeit für die Erledigung einer Aufgabe oder abgeschlossene Aufgaben pro Zeiteinheit [12]. Die Zufriedenstellung wird oft über Fragebögen ermittelt. Die Fragen können offen gestellt sein, meist empfiehlt sich jedoch eine geschlossene Form, bei denen der Nutzer mit Ja/Nein oder einem Punktwert auf einer Bewertungsskala antworten kann. Standardisierte Fragebögen bieten den Vorteil der Objektivität, Reproduzierbarkeit und Quantifizierung [44]. Beispiele für standardisierte Fragebögen sind QUIS (Questionnaire for User Interaction Satisfaction, [7]), SUS (Software Usability Scale, [5]) und PSSUQ (Post-Study System Usability Questionnaire, [32]).

2.2 Mensch-Roboter Interaktion

Mensch-Roboter Interaktion (MRI) ist ein Forschungsbereich, der sich mit dem Verständnis und der Gestaltung von aufeinander bezogenem Handeln zwischen Mensch(en) und Roboter(n) befasst [22]. Ziel der MRI ist die Bildung eines effizienten und effektiven Teams aus Mensch und Roboter, das Nutzen aus den Fähigkeiten beider Partner zieht [49]. Im Gegensatz zur Mensch-Maschine Interaktion befasst sich die MRI mit einem weitaus komplexeren dynamischen Kontrollsystem, das zudem Autonomie und kognitive Fähigkeiten aufweisen kann. Durch die physische Natur des Roboters kommt die Möglichkeit der unmittelbaren Interaktion hinzu, statt eine ausschließliche Interaktion über Mensch-Maschine Schnittstellen.

2.2.1 Klassifikation von Mensch-Roboter Interaktionen

Rolle des Menschen

Scholtz [49] klassifiziert MRI hinsichtlich der Rolle des Menschen. Dieser kann als Aufseher (*supervisor*), Bediener (*operator*), Mechaniker (*mechanic*), Zuschauer (*Bystander*) oder Gleichrangiger (*Peer*) agieren. Der Aufseher überwacht und kontrolliert die Gesamtsituation und evaluiert sie hinsichtlich des Gesamtziels. Er kann Pläne modifizieren und Aktionen spezifizieren. Der Bediener modifiziert Parameter (beispielsweise Kontrollparameter) zur Veränderung des Verhaltens des Roboters oder bedient den Roboter mittels Teleoperation. Der Mechaniker passt die physischen Komponenten des Roboters an, indem er beispielsweise eine Kamera ausrichtet. Zuschauer operieren nicht unmittelbar mit dem Roboter. Sie haben jedoch ein mentales Modell des Roboters und können beispielsweise vorhersagen, ob ein Putzroboter anhält, wenn der Zuschauer in seinem Pfad steht, oder der Zuschauer ihm aus dem Weg gehen muss, um eine Kollision zu vermeiden. Der Gleichrangige kann Anweisungen geben, solange das Gesamtziel weiter verfolgt wird.

Grad der Autonomie

Der Grad der Autonomie kann mit einer fünfstufigen Skala von geringer Autonomie (1) bis hoher Autonomie (5) erfasst werden [16].

1. Roboter liefert keine Unterstützung. Der Mensch muss sämtliche Handlungen selbstständig ausführen.
2. Der Roboter bietet Handlungsempfehlungen an, zwischen denen der Mensch wählen kann.

3. Der Roboter führt Handlungen mit der Zustimmung des Menschen aus.
4. Der Roboter führt Handlungen selbstständig aus, solange der Mensch nicht eingreift.
5. Der Roboter führt Handlungen selbstständig aus und ignoriert den Menschen.

Es gilt zu unterscheiden, welche Funktionen des Mensch-Roboter Systems automatisiert werden. Parasuraman et al. [37] unterscheiden zwischen der Beschaffung von Informationen, der Analyse von Informationen, der Auswahl von Entscheidungen und der Umsetzung von Aktionen. Jede dieser Funktionen kann auf vielfältige Art und Weise automatisiert werden.

Art der Interaktion

MRI kann über unmittelbaren physischen Kontakt zwischen Mensch und Roboter stattfinden (*intimate interaction*) oder mittels einer Schnittstelle, die als Übersetzer zwischen Mensch und Roboter dient (*primitive interaction*). Die Schnittstelle übersetzt die Eingaben des Menschen in Roboterbefehle und gibt Rückmeldungen aus. Sind Mensch und Roboter durch eine räumliche oder zeitliche Distanz getrennt (*loose interaction*), wird die MRI als Teleoperation bezeichnet [56].

Art der Kommunikation

Milgram et al. [34] unterscheiden bei der Kommunikation zwischen Mensch und Roboter zwischen kontinuierlicher und diskreter Sprache. Bei ersterer handelt es sich um Informationen, die kontinuierlich dargeboten werden, wie etwa analoge Anzeigergeräte oder Computermäuse. Diskrete Sprache besteht

aus unabhängigen Elementen, zum Beispiel gesprochene oder geschriebene Befehle sowie Schalter.

2.2.2 Mensch-Roboter Interaktion für die Roboterprogrammierung

Die Roboterprogrammierung ist eine besondere Form der MRI, bei der der Bewegungsablauf eines Roboters zur Erfüllung einer bestimmten Aufgabe definiert wird. Es wird zwischen Online- und Offline-Programmierung unterschieden. Erstere findet direkt am eingeschalteten Roboter statt, während bei der Offline-Programmierung zunächst ohne Roboter programmiert und das Programm anschließend in den Roboter geladen wird. Die Programmierung eines Roboters kann auf verschiedenen Abstraktionsebenen ausgeführt werden. Die niedrigste Ebene ist die Kommandierung von Strömen, gefolgt von der Vorgabe der Winkel der rotatorischen Gelenke des Roboters beziehungsweise der Position der translatorischen Gelenke. Einen höheren Grad der Abstraktion erfordert die Programmierung durch Definition der Position und Orientierung des Endeffektors. Den höchsten Abstraktionsgrad hat die objekt- oder aufgabenbasierte Programmierung, die beispielsweise Befehle wie „Lege den gelben Zylinder auf den roten Würfel ab“ beinhalten könnte. Der Programmierer gibt vor *was* – nicht *wie* – der Roboter etwas ausführen soll. Hierfür muss der Roboter in der Lage sein, seine Umwelt zu modellieren und Aktionen zu planen. [25]

Biggs und MacDonald [3] unterscheiden zwischen manueller und automatischer Roboterprogrammierung. Automatische Roboterprogrammierung gestattet wenig oder keine direkte Kontrolle über den Code, während bei der manuellen Roboterprogrammierung das gewünschte Verhalten direkt spezifiziert wird. In der Regel geschieht dies grafisch oder textbasiert mittels Offline-Programmierung.

Zur automatischen Roboterprogrammierung gehören laut Biggs und MacDonald [3] PdV und instruktive Systeme.

Idee des PdV ist die Ableitung eines Programms durch die Aufnahme einer Sequenz von Aktionen eines Nutzers. Beispielsweise bewegt der Nutzer ein Programmierhandgerät oder den Endeffektor des Roboters, dessen physikalische Bewegung in Form einer Trajektorie in die Programmierung übernommen wird. PdV kann in drei Schritte unterteilt werden: Beobachten, Aktion ableiten und Aktion durchführen. Wird beispielsweise eine Pick&Place Aufgabe vorgemacht, indem der Endeffektor des Roboters zum Objekt am Startort und zum Zielort geführt wird, so besteht der erste Schritt - Beobachten - darin, die Trajektorie in Form der Roboterposen in kleinen Zeitabständen aufzunehmen. Aus dieser Trajektorie kann dann die genaue Aktion einschließlich des Öffnens und Schließens des Greifers sowie gegebenenfalls einer vereinfachten Trajektorie abgeleitet werden. Diese Aktion kann dann abschließend vom Nutzer abgerufen werden.

Bei instruktiven Systemen wird eine Sequenz von Anweisungen vorprogrammiert, die der Nutzer in der Regel durch Gesten- oder Sprachsteuerung abrufen kann.

2.2.3 Mensch-Roboter Schnittstellen

Die Aufgabe von Mensch-Roboter Schnittstellen ist es, den erforderlichen Grad an Bedienumfang für die jeweilige Nutzergruppe zur Verfügung zu stellen. Ein Staubsaugroboter erlaubt beispielsweise auch einem Novizen die Bedienung, während dieser mit der Programmierung eines modernen Industrieroboters sicherlich überfordert wäre.

Fong et al. [18] benennen den Dialog als eines der Kernthemen kollaborativer MRI. Ein Dialog ist der Prozess der Kommunikation zwischen zwei oder mehr Parteien. Er beinhaltet das Teilen von Informationen und die Kontrolle. Bei

der Kommunikation zwischen Mensch und Roboter ist in der Regel eine Schnittstelle involviert, die zur Übersetzung von menschlicher Sprache zu einer Sprache, die der Roboter versteht, und umgekehrt dient. Entscheidend für die Gestaltung einer Mensch-Roboter Schnittstelle ist die Anpassung an die Nutzerzielgruppe. Es müssen die Erfahrungen im Umgang mit robotischen Systemen, die persönlichen Vorlieben, die benötigten Informationen und die Reaktionsbereitschaft in Betracht gezogen werden

Diese Schnittstellen lassen sich wie folgt unterscheiden:

Grafische Benutzerschnittstellen

Grafische Benutzerschnittstellen (engl. GUI) sind in der Roboterprogrammierung weit verbreitet. Das System wird mittels grafischer Symbole und Steuerungselementen bedienbar gemacht. Diese werden auf einem Display angezeigt und mithilfe von Computermäusen, Joysticks oder bei Touchdisplays durch Berührung ausgewählt.

Greifbare Benutzerschnittstellen

Greifbare Benutzerschnittstellen (engl. Tangible User Interface (TUI)) ermöglichen die Interaktion mit einem System durch physische Stellvertreterobjekte. Ein Beispiel für eine solche Schnittstelle ist eine Computermouse, bei der die physische Eigenschaft der Position der Maus auf einer glatten Fläche auf die Position eines Zeigerobjektes auf einem Display übertragen wird.

Sapounidis und Demetriadis [43] haben in einer Studie ein GUI mit einem TUI für die Roboterprogrammierung anhand der Vorlieben und Meinungen von Kindern verglichen. Sie fanden dabei heraus, dass das TUI als attraktiver

wahrgenommen wurde. Insbesondere ältere Kinder, die bereits Erfahrungen im Umgang mit Computern hatten, fanden die Programmierung mithilfe des GUIs jedoch einfacher. TUIs können somit offensichtlich unerfahrenen Nutzern den Programmierprozess erleichtern.

PdV findet häufig mithilfe von TUIs statt, indem zum Vormachen ein greifbares Stellvertreterobjekt genutzt wird.

Natürliche Benutzerschnittstellen

Natürliche Benutzerschnittstellen (engl. Natural User Interface (NUI)) nutzen natürliche Formen der Interaktion, wie zum Beispiel Berührung, Sprache oder Gesten. Seit dem Aufkommen von Smartphones und Tabletcomputern ist insbesondere die Interaktion durch Berührung weit verbreitet. Auch Sprachsteuerung findet einen zunehmenden Eintritt in unseren Alltag. Moderne Smartphones bieten die Möglichkeit über Sprachsteuerung Suchen im Internet durchzuführen, Kurznachrichten zu verfassen oder das aktuelle Wetter zu erfragen. Weniger verbreitet ist bisher die Gestensteuerung. Dabei wird in der Regel mittels einer Kamera die Bewegung des Menschen aufgenommen. Führt dieser eine Geste aus – zum Beispiel eine Wischbewegung oder das Richten des Daumens nach oben – so wird eine definierte Aktion ausgeführt.

Tsarouchi et al. [58] behandeln die Roboterprogrammierung mittels Körper- und Handgesten. Sie definieren ein Vokabular an Gesten, mit dem die Bewegungsrichtung eines Roboters bestimmt werden kann, und zeigen, dass die natürlichen Eingabemethoden für einen Nicht-Experten einfacher zu verstehen sind als herkömmliche Eingabemethoden.

Lauria et al. [30] nutzen Sprachbefehle, um einen mobilen Roboter zu programmieren. Der Roboter versteht Anweisungen wie „Gehe zum Ende der

Straße“, die in ihre syntaktische Struktur zerlegt werden, um nach einer semantischen Analyse ein Programm zur Ansteuerung des Roboters abzuleiten.

Mixed Reality (MR) – gemischte Realität – wird ebenfalls häufig bei der Gestaltung von NUIs eingesetzt. Der Begriff umschreibt die Vermischung von realen und virtuellen Inhalten. Eine spezielle Form der MR - die Augmented Reality - wird in Abschnitt 2.3 genauer behandelt.

Zur Steigerung der Immersion, das heißt der Wahrnehmung der virtuellen Inhalte als realer Bestandteil der Umwelt, werden zusätzlich haptische Schnittstellen eingesetzt. Salisbury et al. [42] beschreiben beispielsweise die Entwicklung einer virtuellen Welt, in der Nutzer mittels zweier haptischer Geräte Würfel im Raum bewegen und deren Reibung, Masse, Trägheit und Stoßwirkung spüren können.

2.2.4 Gebrauchstauglichkeit von Mensch-Roboter Systemen

Scholtz [48] hat folgende Kriterien zur Evaluation von Mensch-Roboter Systemen entwickelt:

1. Sind die notwendigen Informationen vorhanden, sodass der Mensch erkennt, wann sein Eingreifen gefordert ist?
2. Sind die Informationen in einer angemessenen Art und Weise dargestellt?
3. Ist die Interaktionssprache für Mensch und Roboter effizient?
4. Werden Interaktionen effizient und effektiv bewältigt - sowohl aus Sicht des Menschen als auch des Roboters?

Die Darstellung aller notwendigen Informationen in einer angemessenen Art und Weise schafft beim Nutzer Situationsbewusstsein - das Wissen darüber, was um ihn herum geschieht. Dafür muss er ein Verständnis dafür entwickeln, welche Informationen in der aktuellen Situation entscheidend sind. In der MRI muss der Mensch die aktuelle Position des Roboters, seine Aktivitäten, seinen Status und seine Umgebung erfassen. Der Roboter muss sich der Befehle des Menschen und der Einschränkungen seiner Aktivitäten bewusst sein. [13]

Endsley [15] unterscheidet zwischen drei Stufen des Situationsbewusstseins: Wahrnehmung von Informationen (Stufe 1), das Verständnis der Informationen und deren Relevanz in Bezug auf das zu erreichende Ziel (Stufe 2) sowie die Fähigkeit, aus den Informationen und deren Interpretation Vorhersagen über die Zukunft zu treffen (Stufe 3). Ein höherer Grad an Autonomie des Roboters führt zu einem geringeren Situationsbewusstsein beim Nutzer [17, 28]. Desweiteren haben Drury et al. [13] gezeigt, dass kritische Ereignisse bei MRIs auf Verletzungen des Situationsbewusstseins zurückzuführen sind. Kritische Ereignisse sind solche, die einen signifikanten negativen Einfluss auf die Effektivität, Effizienz oder Zufriedenheit des Nutzers mit dem System haben. Bei der Konzeption von MRIs ist daher die Auswahl an Informationen entscheidend, die dem Nutzer präsentiert werden. Dies ist besonders zu beachten, wenn der Roboter Aufgabenteile weitestgehend autonom ausführt und so die Gefahr besteht, dass das Situationsbewusstsein des Nutzers sinkt.

2.3 Augmented Reality

Augmented Reality (AR) – erweiterte Realität – beschreibt innovative Verfahren zur Visualisierung von synthetischen Informationen auf natürlich verständliche Weise [40]. Die menschliche Sinneswahrnehmung wird durch

zusätzliche virtuelle Informationen beeinflusst. Diese werden mittels eines Displays so mit den physikalischen Informationen überlagert, dass der Eindruck entsteht, sie seien Teil der physikalischen Welt [24]. Die Vermischung realer und virtueller Inhalte wird auch als MR bezeichnet. Nach [35] handelt es sich dabei um ein Kontinuum, das sich von der Realität bis zur Virtualität erstreckt (siehe Abbildung 2.1). Überwiegt der Anteil der Realität, spricht man von AR.

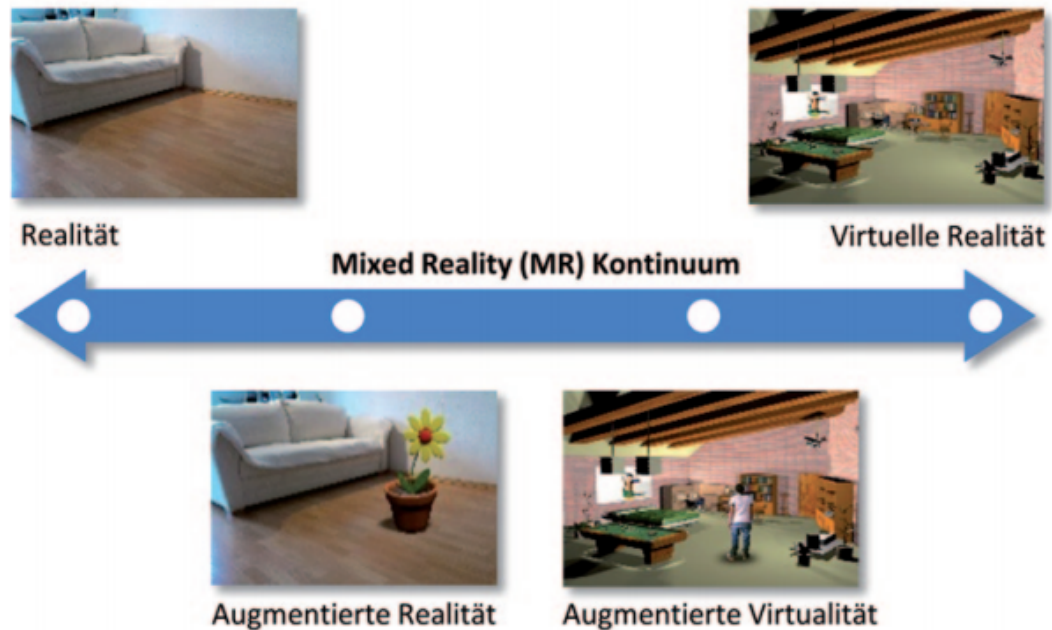


Abb. 2.1.: Mixed Reality Kontinuum nach [35] (aus [14])

AR-Anwendungen nutzen in der Regel Datenbrillen. Desweiteren ist der Einsatz von Projektionen zur Erzeugung von AR-Effekten möglich. Die virtuellen Inhalte werden direkt in die Umgebung projiziert ohne dass es eines Displays bedarf. Ein Beispiel dafür ist das AR-Billard in Abbildung 2.2, bei dem die Positionen der Kugeln sowie des Queues erfasst und die Bewegung der Kugeln auf den Billardtisch projiziert werden.

Im Folgenden werden Begrifflichkeiten aus der AR-Welt erläutert.

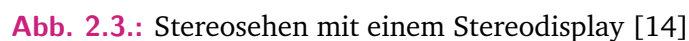


Abb. 2.2.: Erzeugung von AR-Effekten durch Projektion beim Billard [26]

2.3.1 Stereosehen

AR vermittelt dem Menschen, virtuelle Inhalte seien Teil seiner realen Welt, indem sie sich die Eigenschaften der menschlichen Wahrnehmung zunutze machen. Bei der visuellen Wahrnehmung wird Licht über die Linse auf die Retina projiziert. Damit das so entstandene Bild auf der Netzhaut scharf abgebildet wird, stellen Muskeln die Linse je nach Abstand des Objekts entsprechend ein. Auf der Retina befindliche Sehzellen leiten die Informationen an das Gehirn weiter, das sie zur Sinneswahrnehmung des Sehens verarbeitet. Die Informationen des rechten und linken Auges werden dabei zu einem einzigen Bild verschmolzen, das uns einen dreidimensionalen Eindruck unserer Umgebung vermittelt. Das Stereosehen des Menschen führt dazu, dass dieser ein dreidimensionales Objekt auf einem zweidimensionalen Display sieht,

Abbildung 2.3 zeigt, wie das Stereosehen mit einem Stereodisplay umgesetzt wird. Der Punkt P_R ist nur für das rechte Auge sichtbar, während der Punkt P_L nur für das linke Auge sichtbar ist. Das Gehirn interpretiert die Lichtreize so, als ob sie von einem einzigen Lichtpunkt P^* stammen würden. Dieser befindet sich vor der Displayfläche (negative Parallaxe). Scheint die Fusion zweier Lichtpunkte hinter der Displayebene zu liegen, so spricht man von positiver Parallaxe (Q_L , Q_R und Q^*).



Es wird zwischen Optical-See-Through (OST) und Video-See-Through (VST) Geräten unterschieden. Bei ersteren ist das Display semitransparent, sodass der Nutzer die physikalische Welt durch das Display hindurch sieht, während

virtuelle Informationen auf dem Display angezeigt werden. Abbildung 2.4 zeigt den schematischen Aufbau eines OSTs mit semi-transparenten Spiegeln. Es handelt sich um die am weitesten verbreitete Umsetzung von OSTs. Die virtuellen Inhalte werden mittels eines Displays und einer Optik so dargestellt, dass der Betrachter sie anfokusieren kann. Durch die Semitransparenz der Spiegel in der Optik werden sowohl das Umgebungslicht als auch das vom Display stammende Licht abgedunkelt. Die Verschmelzung der virtuellen und physikalischen Informationen findet direkt auf der Netzhaut statt. Im Gegensatz dazu werden bei VST-Geräten die physikalischen Informationen mit einer Kamera aufgenommen und mit den virtuellen Informationen auf einem Display verschmolzen. Der Nutzer betrachtet also nicht direkt die physikalische Welt, sondern das von einer Kamera aufgenommene Bild. [14]

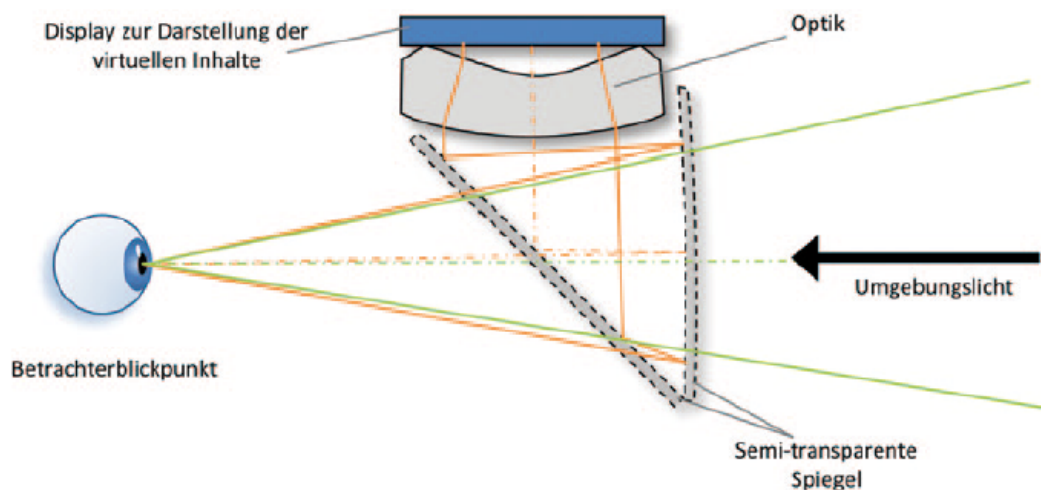


Abb. 2.4.: Schematischer Aufbau eines OSTs (aus [14])

Bei der Darstellung der virtuellen Informationen muss beachtet werden, welche Art von Gerät verwendet wird. Bei OST-Geräten lassen sich beispielsweise dunkle virtuelle Inhalte nicht auf hellem Hintergrund darstellen, da diese transparent sind. VST-Geräte bieten den Vorteil, dass bei Verwendung eines Touchdisplays Interaktionen - beispielsweise die Auswahl eines virtuellen Objekts - durch Berührung des Displays vorgenommen werden können.

Tab. 2.1.: Vergleich von OST-Brillen

| | Gewicht [g] | Sichtfeld [°] | Display [Pixel] | Kamera [Mega Pixel] | Betriebssystem |
|---------------------------|-------------|---------------|-----------------|---------------------|----------------|
| Epson Moverio BT-200 [51] | 88 | 23 | 940 x 540 | 0.3 | Android 4.0 |
| Epson Moverio BT-300 [52] | 69 | 23 | 1280 x 720 | 5 | Android 5.1 |
| Google Glasses [23] | 36 | 14 | 640 x 360 | 5 | Android 4.4 |
| Sony SmartEyeglass [53] | 77 | 20 | 419 x 138 | 3 | Android 4.4 |
| Microsoft HoloLens [33] | 579 | 34 | 1920 x 1200 | 2,4 | Windows 10 |

In Tabelle 2.1 werden die technischen Daten verschiedener OST-Brillen miteinander verglichen. Die Brillen von Epson, Sony und Google haben mit 14° bis 23° ein sehr geringes Sichtfeld. Lediglich die sehr viel größere und schwerere Microsoft HoloLens kommt auf 34°. Die Google Glasses haben ein sehr geringes Gewicht, jedoch sind sie in dieser Liste die einzige Brille mit einem monokularen Display, sodass keine Stereodarstellung möglich ist. Auffallend sind zudem die geringen Auflösungen des Displays der Sony SmartEyeglass - die die Darstellung virtueller Inhalte erheblich einschränkt - und die der Kamera der Epson Moverio BT-200, die der Hersteller damit begründet, dass die Kamera hauptsächlich zur Erkennung von Markern und Codes verwendet wird.

2.3.3 Tracking

Um virtuelle Inhalte mit der physikalischen Welt überlagern zu können, ist eine Schätzung der Position und Lage beziehungsweise Orientierung von Objekten vonnöten. Dieser Vorgang wird als Tracking bezeichnet und liefert eine Transformation des Nutzerkoordinatensystems in das Koordinatensystem der virtuellen Umgebung. Die Qualität und Zuverlässigkeit des Trackings hat entscheidenden Einfluss darauf, ob der Nutzer die virtuellen Informationen

als Teil der physikalischen Welt wahrnimmt. Störeinflüsse wie Verdeckungen und Bildrauschen können leicht zu Diskrepanzen zwischen der virtuellen und physikalischen Welt und damit der Zerstörung der Immersion führen.

Es wird zwischen sensorbasierten und kamerabasierten Verfahren unterschieden. Sensorbasierte Verfahren nutzen Positionssensoren wie GPS, Magnetometer, Beschleunigungssensoren und Gyroskope, um die globale Position des Nutzers in der Umgebung zu bestimmen. Sensorbasierte AR findet beispielsweise bei Applikationen Anwendung, die Sehenswürdigkeiten in der Umgebung auf dem Smartphone-Display anzeigen. Ein großer Nachteil dieses Verfahrens ist die oft hohe Verzögerung der virtuellen Überlagerung bei Bewegungen des Anzeigegeräts [24]. Bei kamerabasierten Verfahren werden Algorithmen zur Erkennung und Verfolgung von Objekten mithilfe des Kamerabilds eingesetzt. Vorteil dieser Verfahren ist die Möglichkeit, auftretende Fehler dynamisch zu korrigieren [2].

Weit verbreitet ist das kamerabasierte Tracking mit Markern. Marker bestehen aus Mustern, die sich auch unter wechselnden Helligkeitsbedingungen leicht aus dem Bild extrahieren lassen. Abbildung 2.5 zeigt ein Beispiel eines solchen Markers. Diese Marker werden unmittelbar auf oder an einem Objekt angebracht, an dem virtuelle Inhalte angezeigt werden sollen. Für das Tracking mit Markern wird im Kamerabild die Kontur des Markers mittels Linienextraktion ermittelt. Anhand der Position der Konturpunkte können daraus dann Position und Lage der Kamera relativ zum Marker berechnet werden. Neben dem Tracking mit Markern kommen auch merkmalsbasierte Tracking-Verfahren zum Einsatz. Es werden Merkmale im Kamerabild extrahiert und Modellen in einer Datenbank zugeordnet. Merkmale können sowohl Kanten und Eckpunkte als auch Deskriptoren (individuelle Beschreibungen) von Bildpunkten sein. Die Verwendung von Kanten- und Eckpunkten bietet sich insbesondere dann an, wenn wenig andere Merkmale zur Extraktion vor-

handen sind. Die Geschwindigkeit und Zuverlässigkeit des merkmalsbasierten Trackings hängt stark von den verwendeten Deskriptoren ab. [14]



Abb. 2.5.: Beispiel für einen Marker zum kamerabasierten Tracking [39]

2.3.4 Registrierung

Geometrische Registrierung ist die feste Verortung virtueller und realer Inhalte, indem deren Koordinatensysteme auf Basis der Positions- und Lageschätzung des Trackings in Beziehung gebracht werden.

2.3.5 Rendering

Unter Rendering versteht man die Darstellung virtueller Inhalte, die perspektivisch an die reale Welt angepasst werden, sodass diese als Einheit erscheinen. In der vorliegenden Arbeit wird zum Rendern von 2D- und 3D-Objekten das plattformunabhängige Application Programming Interface (API) OpenGL genutzt. Als Beispiel des Renderns eines 3D-Objektes wird im Folgenden daher die Renderingpipeline von OpenGL herangezogen, die in Abbildung 2.6 dargestellt ist.

In den Darstellungslisten werden einmalig bei der Definition alle Eckpunkt- und Pixeldaten gespeichert. Die Listen können wiederholt aufgerufen werden und ermöglichen die schnelle Darstellung statischer Daten, ohne dass

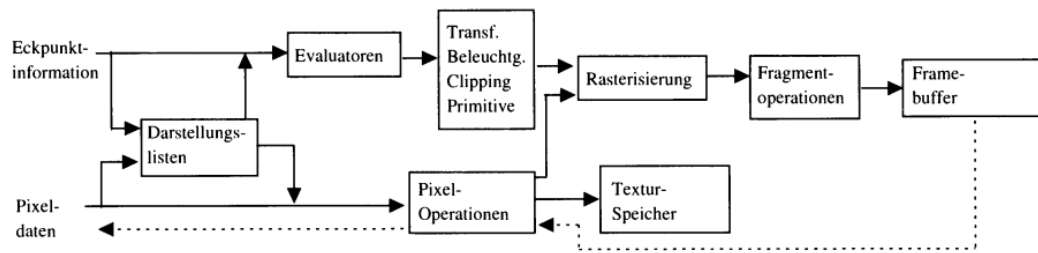


Abb. 2.6.: Renderingpipeline in OpenGL [9]

eine große Menge an Daten bei der Zeichnung eines jeden Frames ausgetauscht werden muss. Die Eckpunktdaten werden transformiert. Dazu wird das Objektkoordinatensystem über das Kamerakoordinatensystem, das Clippingkoordinatensystem und die normierten Projektionskoordinaten in die Bildschirmkoordinaten überführt. OpenGL nutzt 4×4 Matrizen zur Transformation. Optional werden Beleuchtungsberechnungen vorgenommen, die die Farbwerte der Eckpunkte beeinflussen. Zudem können mittels Culling Rückflächen von Polygonen verworfen werden, die ohnehin nicht sichtbar wären. Nachdem die Pixeldaten ausgelesen wurden, werden Operationen wie Skalieren, Mappen und Clamping durchgeführt, bevor die Daten im Texturspeicher gespeichert oder direkt rasterisiert werden. Bei der Rasterisierung werden aus den Pixel- und Eckpunktdaten Fragmente erstellt. Die Eckpunkte werden zu Linien oder Polygonen verbunden und auf das 2D-Punktgitter projiziert. Fragmente sind rechteckige Matrizen, die Farb- und Tiefenwerte sowie Linienbreite, Punktgröße und Antialiasing-Berechnungen beinhalten. Nach der Rasterisierung finden weitere Operationen mit den Fragmenten statt. Texturelemente werden aus dem Texturspeicher generiert und auf jedes Fragment angewendet, Überdeckungen werden berechnet. Die so entstandenen Pixeldaten werden abschließend in den Framebuffer geladen.

2.3.6 Anwendungen in der Mensch-Roboter Interaktion

Bei Lambrecht und Krüger [29] kann der Nutzer Posen und Trajektorien mittels Gestensteuerung definieren. Die Erfassung der Gesten sowie die grafische Darstellung der Posen und Trajektorien findet in einer AR-Umgebung mit einem VST-Gerät statt. Gaschler et al. [20] nutzen ebenfalls ein VST-Gerät sowie ein spezielles 3D-Eingabegerät, um Posen zu definieren und Pfade um virtuelle Objekte zu planen. Der Einsatz von AR-Laserprojektion in der intuitiven Roboterprogrammierung ist in [62] beschrieben. Sie wird dazu genutzt, Trajektorien und Zielkoordinaten in die Umgebung des Roboters zu projizieren. Bei Akan et al. [1] kann der Nutzer in einer AR-Umgebung auf einem VST-Gerät Objekte für eine Pick&Place-Aufgabe auswählen und erhält visuelle Rückmeldungen während des Vorgangs. Chong et al. [8] lassen den Nutzer mittels einer AR-Umgebung den kollisionsfreien Raum sowie Start- und Ziel-Konfiguration definieren, um daraus den Pfad des Roboters zu berechnen. Ein OST-Gerät findet in [38] Anwendung. Der Nutzer kann mittels eines tragbaren Eingabegeräts Punkte zur Pfadplanung für eine Lackieraufgabe definieren. Eine grafische Repräsentation der Lackierung wird mittels AR unmittelbar auf das Zielobjekt projiziert.

2.3.7 Probleme bei Verwendung von Optical-See-Through-Geräten

Da in dieser Arbeit auf OST-Geräte zurückgegriffen wird, werden im Folgenden verschiedene Probleme erläutert, die sich im Umgang mit AR auf OST-Geräten ergeben und zu einer Beeinträchtigung der Gebrauchstauglichkeit führen können. Es ist dabei zu unterscheiden, ob die Probleme durch die

Hardware, die Software oder die grundsätzliche Konzeptionierung bedingt sind.

Hardware

OST-Geräte sind in der Regel als Datenbrillen realisiert, die mit verschiedenen Sensoren sowie ein (monokular) oder zwei (binokular) Displays ausgestattet sind. Durch die benötigten Technikkomponenten sind die Datenbrillen meist relativ groß und schwer. Die in dieser Arbeit verwendete Epson Moverio BT-200 wiegt beispielsweise 88 Gramm und misst 170 mm mal 185 mm mal 32 mm. Sie ist damit deutlich größer und schwerer als eine gewöhnliche Brille zur Korrektur der Sehkraft. Hinzu kommt, dass Brillenträger ihre Brille zusätzlich zur Datenbrille tragen müssen. Des Weiteren ist die oben genannte Brille über ein Kabel mit einem Controller verbunden, der wiederum bei der Kommunikation über USB mit einem PC verbunden ist, sodass die Bewegungsfreiheit eingeschränkt ist. Ebenso ist das Sichtfeld mit 23° relativ gering. Schaut der Nutzer beispielsweise aus fünf Metern auf eine Wand, so hat der Bildschirm lediglich eine Diagonale von etwa zwei Metern. Die virtuellen Inhalte können auf den Displays nicht beliebig hell dargestellt werden, sodass sich eine schlechte Sichtbarkeit bei hellem Hintergrund oder durch Reflexionen ergeben kann. Für eine störungsfreie Verortung der virtuellen Inhalte sind zudem die Fähigkeiten der Sensoren (Kamera, Inertial Measurement Unit (IMU)) entscheidend. Eine Kamera mit einer geringen Auflösung erschwert beispielsweise die Detektion von Markern.

Software

Es finden verschiedene Verarbeitungsschritte statt, die Latenzen verursachen. Bei einer hohen Latenz bewegen sich die virtuellen Inhalte bei Bewegungen

des Kopfes mit und haben einen Versatz bezüglich der korrekten Position. Die Latenzen sorgen somit für eine Verringerung der Immersion. Die Gesamtlatenz setzt sich aus der Tracking-, Transport-, Simulations-, Generierungs- und Darstellungslatenz zusammen [14]. Die Trackinglatenz tritt zwischen der Bewegung des Nutzers und der Verfügbarkeit der Bewegungsinformationen auf. Findet die Verarbeitung beispielsweise auf einem externen Computer statt, sodass die Daten zunächst übertragen werden müssen, kann es zusätzlich zu Transportlatenzen kommen. Bei der Berechnung der Posen der Marker in der realen Welt können Simulationslatenzen auftreten. Generierungslatenzen beschreiben Verzögerungen durch das Rendering der virtuellen Inhalte, die anschließend an das Ausgabegerät übermittelt werden müssen, sodass Darstellungslatenzen auftreten können.

Desweiteren beeinflusst die Tracking-Rate (Anzahl der Tracking-Ergebnisse pro Sekunde) die Qualität der Registrierung. Ist die Tracking-Rate zu gering, bewegen sich die virtuellen Inhalte bei Bewegungen des Kopfes mit diesen kurzzeitig mit, bevor sie auf ihre Position in der realen Welt zurückspringen.

Konzeptionierung

Der dreidimensionale Gestaltungsraum in der AR hat eine höhere Anzahl an Freiheitsgraden im Gegensatz zu herkömmlichen zweidimensionalen GUIs. Dadurch lassen sich die etablierten und erprobten Interaktionsparadigmen von GUIs nicht ohne Weiteres auf AR-Interaktionen übertragen.

Diskrepanzen zwischen der wirklichen und den simulierten Bewegungen können zur sogenannten Cybersickness führen. Die visuellen Informationen stimmen nicht mit den vestibulären Informationen überein, sodass es beim Nutzer zu Benommenheit, Schwindelgefühlen und Übelkeit kommen kann. [31]

Ist der Unterschied zwischen den beiden Bildern, die vor dem rechten und dem linken Auge dargestellt werden, zu groß, kann das Gehirn die Bilder nicht fusionieren und der Nutzer nimmt die virtuellen Inhalte als Doppelbilder wahr [4]. Zur Vermeidung des Auftretens von Doppelbildern ist es wichtig, die Datenbrille für jeden Nutzer individuell zu kalibrieren. Zudem kann das Auftreten von Doppelbildern verhindert werden, indem die virtuellen Objekte näher an das Display herangebracht oder mit weniger Details dargestellt werden [14].

2.3.8 Gestaltungsrichtlinien

Richtlinien zur Gestaltung von AR-Anwendungen beziehen sich in der Regel auf spezielle Anwendungsgebiete, zum Beispiel Spiele [59], Anleitungen [41] oder Anwendungen auf mobilen Smartphones und Tabletcomputern [19]. Es lassen sich jedoch daraus folgende allgemeinen Richtlinien ableiten:

- Textuelle Informationen werden klar dargestellt.
- Es besteht ausreichender Kontrast zwischen virtuellen Inhalten und Hintergrund.
- Zusammengehörige Inhalte werden gruppiert.
- Es werden keine wichtigen Informationen in der realen Welt durch virtuelle Inhalte verdeckt.
- Der Nutzer ist in der Lage wichtige Informationen, die seine Aufmerksamkeit erfordern, zu identifizieren.
- Die virtuellen Inhalte werden einfach gehalten.

2.4 Roboterkinematik

Ein Roboter besteht aus Gliedern, die in einer Kette durch Gelenke verbunden sind. Der unbewegliche Teil des Roboters wird als Basis bezeichnet. Am freien Ende der Kette befindet sich der Endeffektor, beispielsweise ein Greifer oder Bohrer. Die Anzahl der Gelenke bestimmt die Freiheitsgrade eines Roboters. Die Position und Orientierung der Basis, der Gelenke und des Endeffektors zueinander können durch kinematische Gleichungen beschrieben werden. Die Beziehung zwischen dem Basis- (B) und Endeffektorkoordinatensystem (E) des 3 Degrees of Freedom (DOF)-Roboters in Abbildung 2.7 kann beispielsweise mittels der Transformationsmatrix

$${}^B_E T = \begin{pmatrix} {}^B_E R & {}^B P_{E,org} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$

beschrieben werden, die sich aus der Rotationsmatrix ${}^B_E R$ und der Translation ${}^B P_{E,org}$ vom Ursprung des Basis- zum Ursprung des Endeffektorkoordinatensystems zusammensetzt.

Zur Festlegung der Koordinatensysteme gilt folgende Konvention [10]:

1. Die Gelenkachsen werden identifiziert und benachbarte Achsen mit einer Geraden verbunden.
2. Die Senkrechten zwischen den in Schritt 1 gezeichneten Geraden beziehungsweise deren Schnittpunkte werden identifiziert. Der Ursprung der Koordinatensysteme ist der Schnittpunkt beziehungsweise der Punkt, an dem die Senkrechte die Gelenkachse schneidet.
3. Die z-Achse zeigt in Richtung der Gelenkachse.

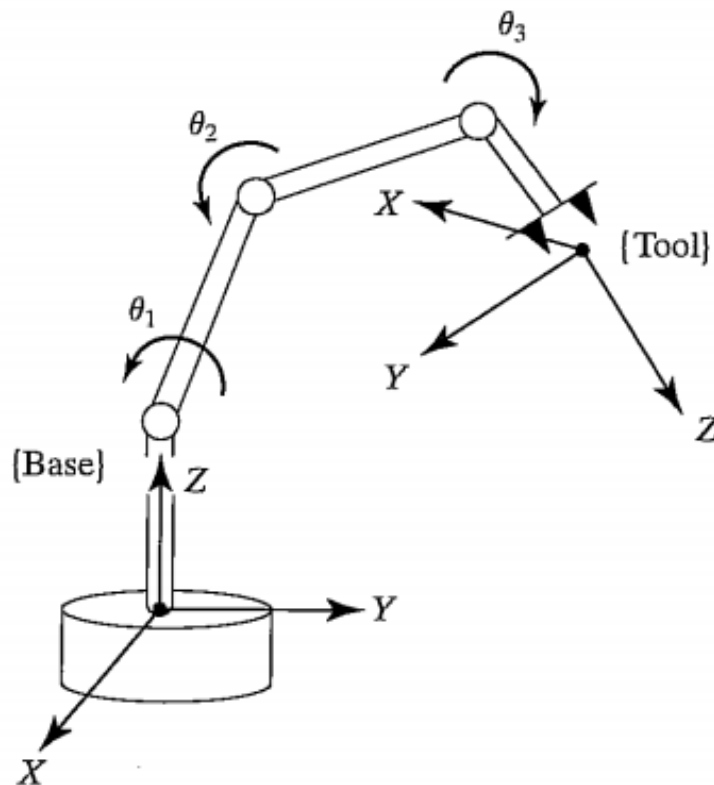


Abb. 2.7.: Beispiel eines 3DOF-Roboters [10]

4. Die x-Achse zeigt in Richtung der Senkrechten oder - falls sich die Achsen schneiden - in Richtung der Normalen der Ebene in der beide Achsen liegen.
5. Die y-Achse wird so festgelegt, dass ein rechtshändiges Koordinatensystem entsteht.
6. Das Basiskoordinatensystem wird auf das erste Gelenk gelegt, wenn die Gelenkvariable null ist. Position und Orientierung des Endeffektorkoordinatensystems werden so gewählt, dass möglichst viele Gliedparameter null sind.

Methodenteil

3.1 Technische Rahmenbedingungen

3.1.1 RAZER

Das in dieser Arbeit entwickelte Interaktionskonzept ist Bestandteil von RAZER, einem Framework zur intuitiven Roboterprogrammierung im Fertigungskontext, das am DLR in Oberpfaffenhofen entwickelt wurde [54]. Das Framework ermöglicht die aufgabenbasierte Online-Programmierung eines oder mehrerer Roboter. Experten definieren generische Skills, wie zum Beispiel *Bohren* oder *Bewegen*, und die nötigen Parameter. Über eine intuitive grafische Oberfläche kann ein Nutzer die Skills anordnen und parametrisieren, um sie an die Aufgabenstellung anzupassen. Posen und Trajektorien werden durch Vormachen definiert, indem der Roboter mit Gravitationskompensation vom Nutzer bewegt wird. Die so entstandene Sequenz an Skills wird als Task bezeichnet. Dieser kann anschließend ebenfalls über die grafische Oberfläche ausgeführt und überwacht werden.

Abbildung 3.1 zeigt einen Überblick über die verschiedenen Komponenten von RAZER. Die definierten Skills mitsamt ihrer Parameter sind in einer Datenbank gespeichert. Das Frontend liest die Skills aus und führt den Nutzer durch die Parametrisierung. Dazu wird zunächst der Roboter, der den Skill durchführen soll, und anschließend der Skill selbst ausgewählt. Optional kann eine Beschreibung angegeben werden, die dann in der Skillübersicht erscheint, um etwa mehrere *Bohren*-Skills unterscheiden zu können (siehe Abbildung 3.2). Anschließend kann der Nutzer die Parameter des Skills be-

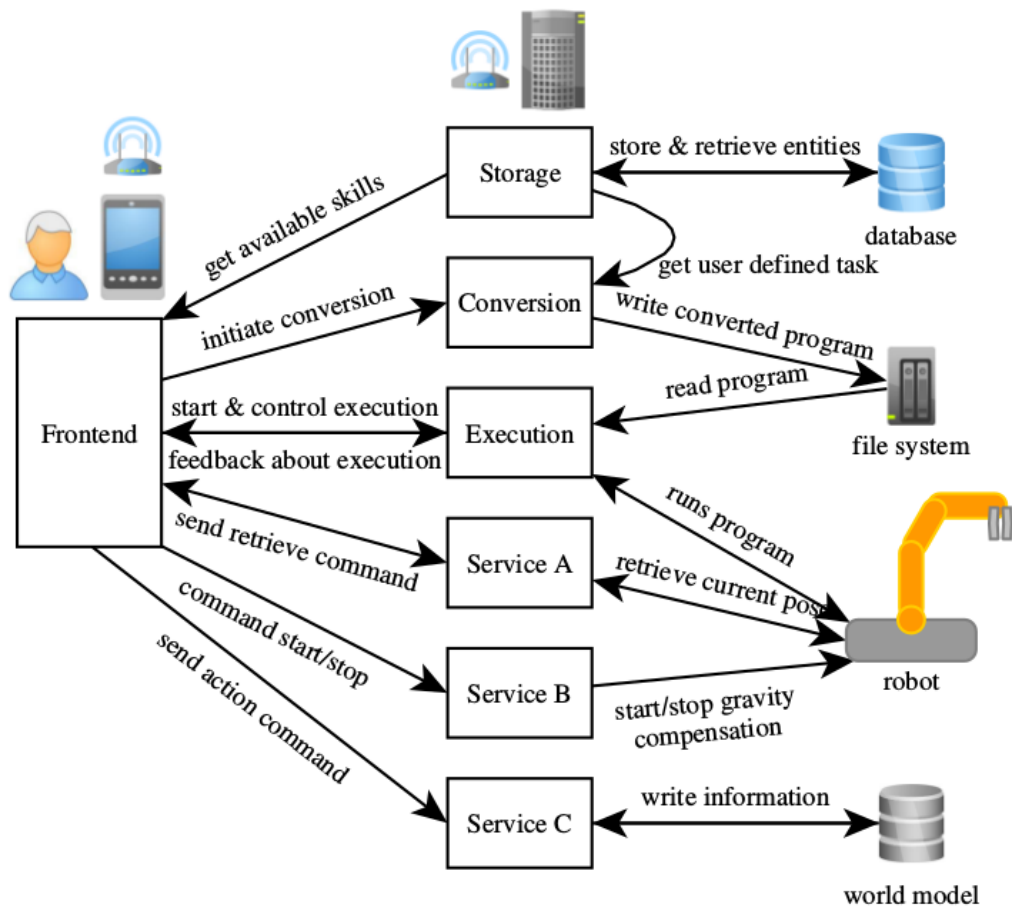


Abb. 3.1.: Überblick über die Komponenten von RAZER [55]

stimmen. Beim *Bohren* könnten dies beispielsweise der Durchmesser des Bohrers, das Material und die Bohrtiefe sein. Je nach Skill kann ein Muster festgelegt werden. Hier wählt der Nutzer aus, ob eine einfache Pose, mehrere (unabhängige) Posen, ein Linienmuster oder ein Gittermuster verwendet werden sollen. Anschließend leitet ein Dialog durch den Vorgang des Vormachens, bei dem der Nutzer den Endeffektor des Roboterarms in die gewünschte(n) Pose(n) bewegt.

Die parametrisierten Skills werden dem Nutzer anschließend als Sequenz - dem Task - angezeigt (siehe Abbildung 3.3). Der Nutzer hat die Möglichkeit, die Reihenfolge zu ändern oder einzelne Skills zu löschen oder zu modifizieren. Der Task kann nun entweder für eine spätere Verwendung abgespeichert oder ausgeführt werden. Dazu wird er in einen Zustandsautomaten [6] über-

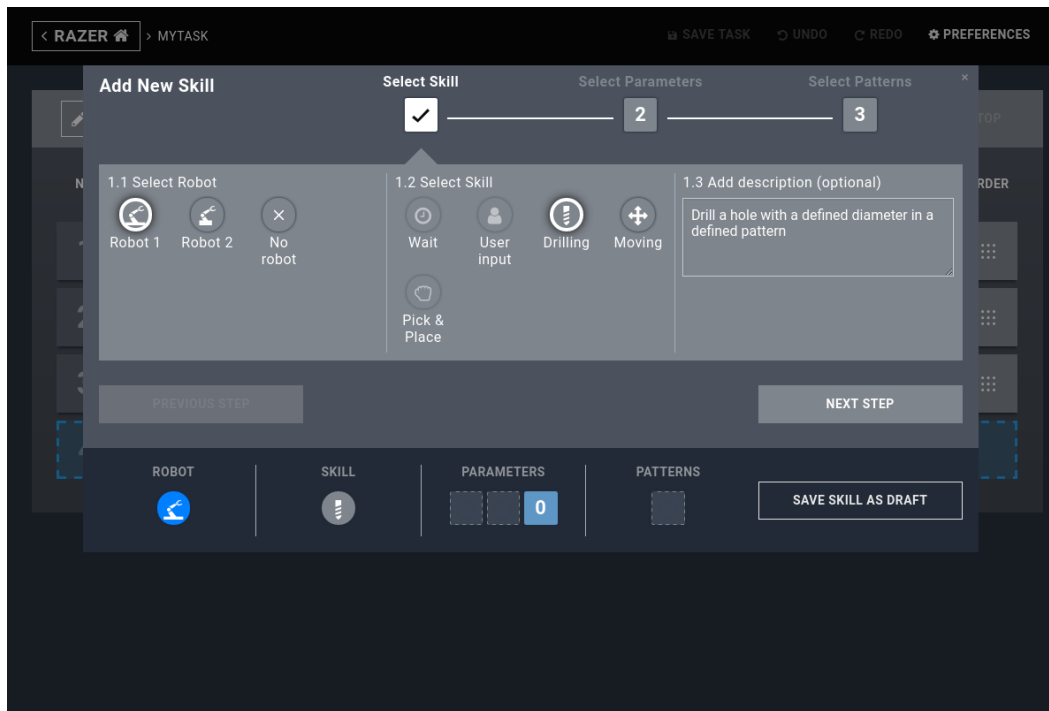


Abb. 3.2.: GUI bei der Erstellung eines neuen Skills

führt, dessen Ausführung vom Frontend aus gestartet und überwacht werden kann.

Die Ansteuerung und Kommunikation mit den Robotern findet über die institutseigene Software *Links&Nodes* statt. Diese erlaubt die Erstellung und Wartung verteilter Computer-Netzwerke. Funktionen zum Öffnen und Schließen des Greifers, der Wechsel des Roboters in den Gravitationskompensationsmodus oder die Aufnahme der Trajektorie während einer Bewegung ist mithilfe von Webservices mit einem Representational State Transfer (REST)-Interface umgesetzt.

3.1.2 Datenbrille

Als Datenbrille wird eine Epson Moverio BT-200 verwendet (siehe Abbildung 3.4). Die Brille ist mit Kamera, GPS sowie Magnetometer, Beschleunigungssensor und Gyroskop ausgestattet. Sie wird an eine Kontrolleinheit

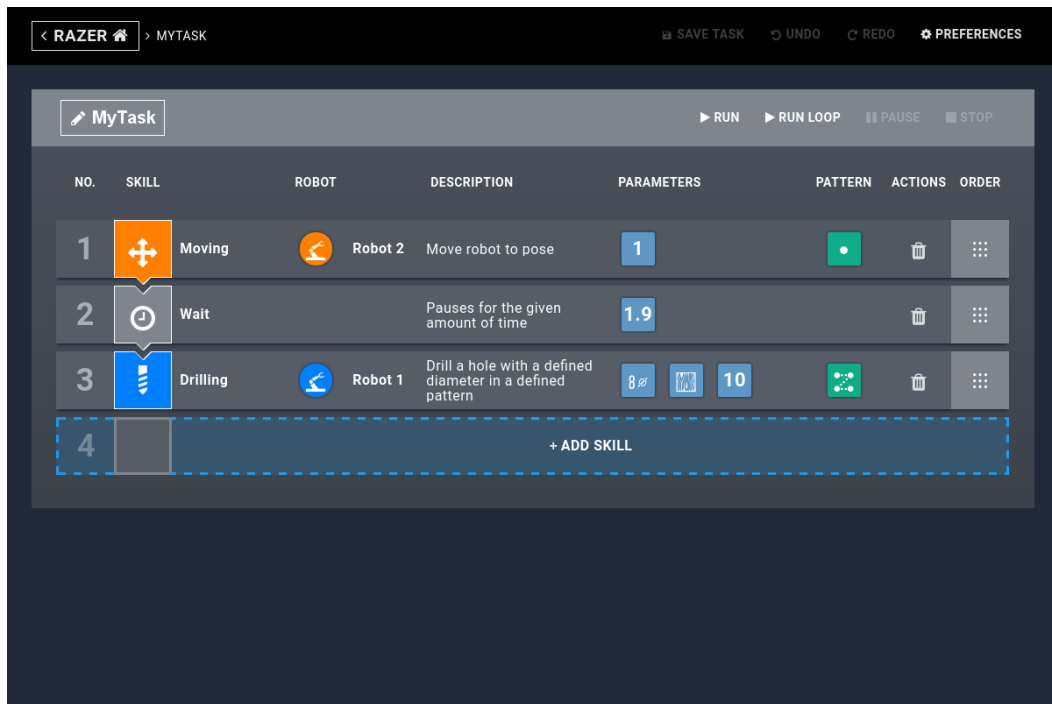


Abb. 3.3.: Sequenz aus drei parametrisierten Skills

angeschlossen, die über ein Touchpad verfügt und Kommunikation mittels WLAN, USB oder Bluetooth erlaubt. Die Datenbrille ist mit Android 4.0 ausgestattet. Weitere technische Daten können Tabelle 2.1 entnommen werden.

Die Programmierung der Datenbrille findet über die Integrated Development Environment (IDE) *Android Studio* statt.

3.1.3 Roboter

Zur Interaktion wird ein 7-DOF KUKA LWR verwendet, der auf einer Linearachse befestigt ist. Ursprung des Basiskoordinatensystems (Koordinatensystem 0), ist der Nullpunkt der Linearachse. In Abbildung 3.5 ist die Gelenkkonfiguration des Roboterarms nach der Konvention in Abschnitt 2.4 abgebildet. Die Koordinatensysteme werden im folgenden von 1 bis 7 durchnummeriert, das Endeffektorkoordinatensystem hat die Nummer 7. Die Marker zur Verortung der virtuellen Inhalte werden statisch in Bezug auf Koordinatensystem



Abb. 3.4.: Epson Moverio BT-200 Smart Glasses [51]

6 angebracht. Da die Transformation 0_7T von der Basis zum Endeffektor ausgelesen werden kann, wird lediglich die Transformation 6_7T benötigt, um in Bezug zu Koordinatensystem 6 die virtuellen Inhalte anzuzeigen und die externen Kräfte auswerten zu können. Da die Ursprünge der Koordinatensysteme 6 und 7 identisch sind, ist Transformation ausschließlich vom Gelenkparameter θ_6 abhängig. Es gilt

$${}^6_7T = \begin{pmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

Die benötigte Transformation 0_6T kann anschließend folgendermaßen berechnet werden

$${}^0_6T = {}^0_7T \cdot {}^6_7T^{-1} \quad (3.2)$$

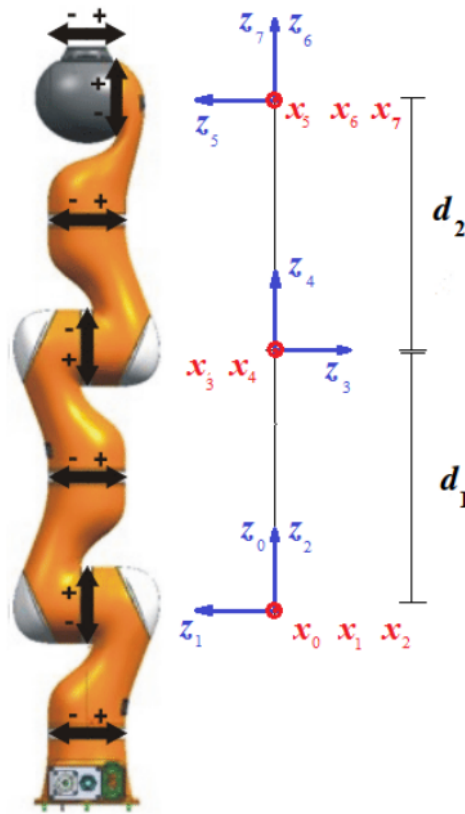


Abb. 3.5.: Gelenkconfiguration des KUKA LWR. Die x-Achsen zeigen in Richtung des Lesers. [21]

3.2 Konzeption der Mensch-Roboter-Schnittstelle

3.2.1 Nutzungskontext

Die Zielnutzer des Systems sind Arbeiter in der Fertigung. Sie verfügen über wenig oder keine Erfahrung in der Roboterprogrammierung, kennen jedoch den Fertigungsablauf. Es sind große Varianzen in den Fertigkeiten im Umgang mit grafischen Oberflächen auf Tabletcomputern zu erwarten.

Der Nutzer kann lesen und verfügt über normale Sehfertigkeiten sowie eine normale manuelle Geschicklichkeit. Arbeitsaufgabe ist, die Programmierung einer oder mehrerer Roboter für einen Fertigungsablauf durchzuführen. Dazu müssen die einzelnen Schritte des Fertigungsablaufs zusammengestellt und parametrisiert werden. Es werden eine grafische Benutzerschnittstelle auf einem Tabletcomputer, eine Datenbrille und direkte Interaktion mit den Robotern genutzt. Durch den Einsatz in der Fertigung kann die Umgebung laut, staubig, schlecht beleuchtet und/oder warm sein.

3.2.2 Systemanforderungen und -design

Im Folgenden werden die Anforderungen an das System zur Lösung des in Abschnitt 1.1 beschriebenen Problems aufgestellt und deren Umsetzung beschrieben.

Funktionale Anforderungen

Der Nutzer hat die Möglichkeit, Eingaben zu tätigen, während er mit dem Roboter interagiert. Dem Nutzer wird eine Auswahl an kontextabhängigen Aktionen über eine AR-Datenbrille am Endeffektor des Roboters angezeigt. Die Auswahl und Betätigung einer Aktion erfolgt durch das Drücken des Roboterarms in die angezeigte Richtung. Befindet sich der Roboter im Gravitationskompensationsmodus, kann der Nutzer Aktionen mit einem Kopfnicken bestätigen und mit einem Kopfschütteln abbrechen.

Qualitätsanforderungen

In [27] werden Qualitätsanforderungen an System- und Softwarequalität aufgeführt. Dazu gehören Funktionalität, Effizienz, Kompatibilität, Benutzbarkeit, Zuverlässigkeit, Sicherheit, Wartbarkeit und Übertragbarkeit.

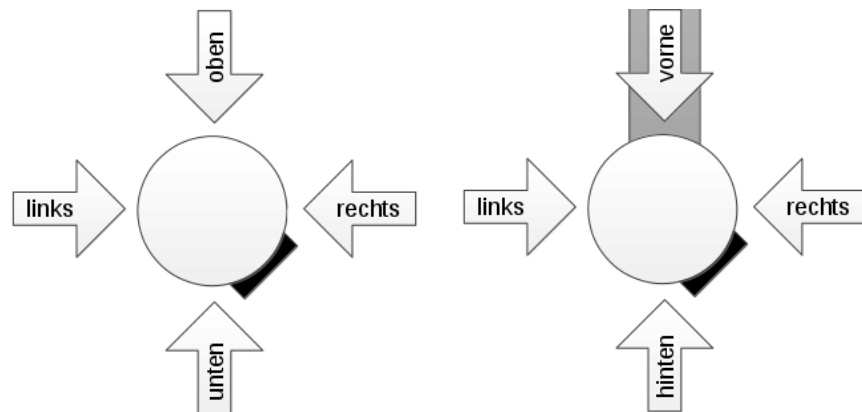
Systemdesign

Der Roboterarm selbst dient als Eingabegerät, indem das Gelenk 6 (siehe 3.5) nach oben, unten, rechts, links, vorne oder hinten gedrückt wird. Zur Definition der Eingaberichtungen in Abhängigkeit von der Pose des Roboters sind zwei Eingabekonzepte möglich:

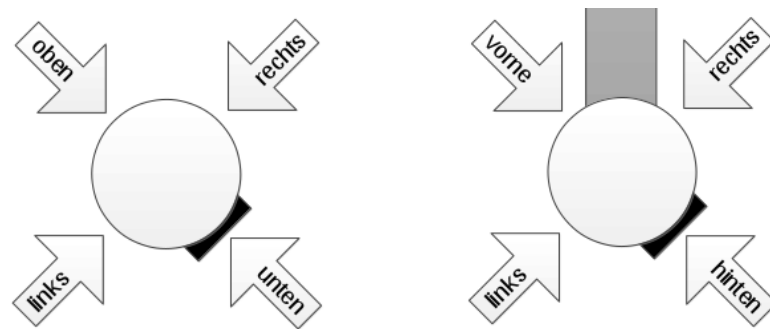
Endeffektororientierte Eingabe Die Orientierung des Endeffektors bestimmt die Richtung, in der Eingaben getätigt werden können. *Oben* und *unten* werden durch Kräfte in Richtung der z-Achse des Endeffektors registriert, *rechts* und *links* durch Kräfte in Richtung der x-Achse und *vorne* und *hinten* durch Kräfte in Richtung der y-Achse (siehe 3.6(c) und 3.6(d)).

Basisorientierte Eingabe Die Eingaberichtungen *oben* und *unten* sind unabhängig von der Position des Endeffektors. *Oben* ist stets der Druck in Richtung der Gravitation, *unten* stets entgegen der Gravitation. Dies ist in 3.6(a) abgebildet. Auch wenn der Endeffektor in eine andere Richtung zeigt, verändern sich die Eingaberichtungen *oben* und *unten* nie. Die Eingaberichtungen *rechts*, *links*, *vorne* und *hinten* werden durch den Winkel des Endeffektors zur Basis in der x-y-Ebene des Basiskoordinatensystems bestimmt. Steht der Nutzer so, dass sich der Endeffektor genau zwischen ihm und der Basis des Roboters befindet, werden mit einem seitlichen Druck die Eingaberichtungen *rechts* und *links* aktiviert.

Der Druck vom Nutzer weg zur Basis hin ist *hinten* und das Ziehen des Gelenks zum Nutzer hin ist *vorne* (siehe 3.6(b)).



(a) Frontalansicht des basisorientierten Eingabekonzepts (b) Draufsicht des basisorientierten Eingabekonzepts



(c) Frontalansicht des endeffektororientierten Eingabekonzepts (d) Draufsicht des endeffektororientierten Eingabekonzepts

Abb. 3.6.: Basisorientiertes und eingabeorientiertes Eingabekonzept am Roboter-gelenk (Kreis). Die schwarze Markierung symbolisiert den Endeffektor und die graue Markierung die Richtung zur Basis.

Grundsätzlich wäre die Implementierung beider Interaktionskonzepte wünschenswert, sodass der Nutzer das ihm intuitiver erscheinende Konzept auswählen und verwenden kann. Aufgrund des begrenzten Zeitrahmens dieser Arbeit wird im Folgenden ausschließlich das basisorientierte Eingabekonzept betrachtet.

Damit der Nutzer weiß, welche Aktion er durch die Wahl der Eingaberichtung auslöst, ist eine grafische Anzeige vonnöten. Diese wird mithilfe einer AR-Datenbrille realisiert, die dem Nutzer die nötigen Informationen anzeigt. Die Darstellung wurde in einem 3D-Grafikprogramm entworfen. Abbildung 3.7

zeigt zwei dieser Entwürfe für vier Eingaberichtungen. Entwurf 3.7(a) besteht aus pfeilförmigen Objekten, auf deren runden Seiten das Icon der Aktion angezeigt werden kann. Die Pfeilspitze zeigt in die Richtung, in die das Roboter-gelenk gedrückt werden muss, um die entsprechende Aktion auszulösen. Entwurf 3.7(b) nutzt zylinderförmige Objekte auf denen ebenfalls das Icon der Aktion angezeigt werden kann. 3D-Hände deuten die Eingaberichtung an.

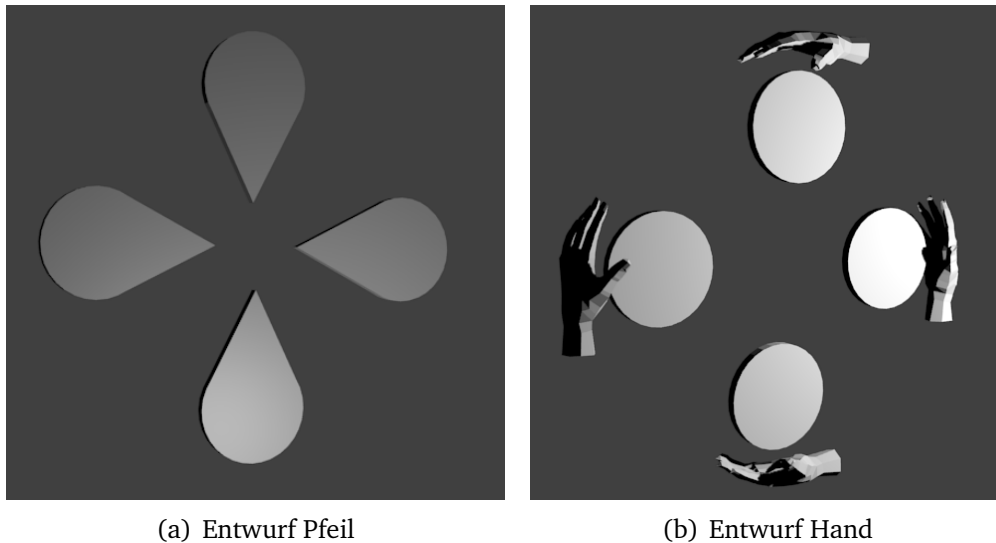


Abb. 3.7.: Entwürfe der grafischen Anzeige für das Interaktionskonzept

Vorteil des Entwurfs 3.7(a) ist die minimalistische Darstellung, sodass der Nutzer nicht durch eine überladene Anzeige überfordert wird. Nachteilig ist, dass unter bestimmten Sichtwinkeln die Icons auf den Pfeilen nicht sichtbar sind. Da die Zylinder in Entwurf 3.7(b) keine Richtungsinformationen liefern, können sie stets auf den Nutzer ausgerichtet werden, sodass der Sichtwinkel hier keinen Einfluss auf die Sichtbarkeit der Icons hat. Jedoch können die virtuellen Hände zu Verdeckungen führen. Dies kann durch eine semi-transparente Darstellung vermindert werden. In Entwurf 3.7(b) wird dem Nutzer zudem bildlich dargestellt, wie er eine Aktion durchführen kann. Dies erleichtert insbesondere Erstbenutzern die Bedienung. Aus diesen Gründen fiel die Entscheidung für die Umsetzung des Entwurfs 3.7(b) aus.

Abbildung 3.8 zeigt die Komponenten, die zur Umsetzung des AR-Interaktionskonzepts implementiert werden. Herzstück ist der *Augmented-Button-Service (ABService)*, der über den *Input-Output-Service (IOService)* mit dem Frontend kommuniziert, Roboterdaten ausliest und alle nötigen Informationen zur Anzeige an die Datenbrille sendet.

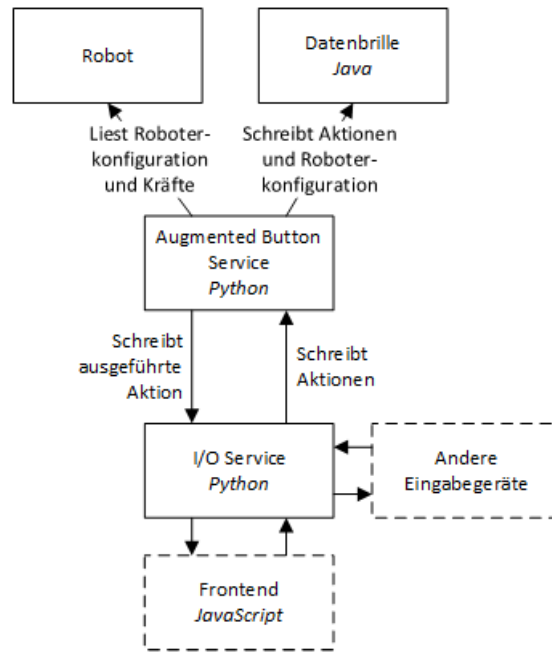


Abb. 3.8.: Komponenten des AR-Interaktionskonzepts

3.2.3 Komponentenanforderungen und -design

Die Anforderungen an die Komponenten sowie deren Umsetzung wird im Folgenden beschrieben.

Schnittstellen

Um die Kommunikation zwischen den Komponenten zu gewährleisten, werden folgende Schnittstellen definiert:

IOService ↔ **Frontend** mittels WebSocket-Protokoll¹. Das Frontend sendet ein Zustandsobjekt (bestehend aus Zustandsbeschreibung mit Aktionen und Gruppen) und der IOService sendet ein Aktionsobjekt.

IOService ↔ **ABService** mittels *Links&Nodes*. Der IOService sendet ein Zustandsobjekt, der ABService sendet ein Aktionsobjekt.

ABService → **Roboter** mittels *Links&Nodes*. Der ABService liest die Roboterparameter (Roboterpose, Gelenkkonfiguration und externe Kräfte) aus.

ABService ↔ **Datenbrille** mittels Android Open Accessory (AOA)-Protokoll. Der ABService sendet Markerorientierung und Aktionsobjekte an die Datenbrille. Die Datenbrille sendet, ob ein Nicken oder Kopfschütteln registriert wurde.

Input-Output-Service

Anforderungen

- Es können sich Ein- und Ausgabemodule beim IOService registrieren.
- Ausgabemodule können sämtliche verfügbare Eingaben, die zu einer Zustandsänderung des Moduls führen, an den IOService übermitteln.
- Eingabemodule werden vom IOService informiert, wenn sich die verfügbaren Eingaben ändern.
- Erfolgt eine Eingabe, so informiert das Eingabemodul den IOService, der wiederum das zugehörige Ausgabemodul benachrichtigt.

Umsetzung Es wird zunächst das Frontend modifiziert, das als Ausgabemodul fungiert. Dazu wird ein Zustandsautomat implementiert, in dem jede Ansicht im Frontend einem Zustand entspricht. Jeder Zustand besitzt ein

¹Das WebSocket-Protokoll erlaubt bidirektionale Kommunikation zwischen einer Webanwendung und einem Webserver.

Eingabealphabet, dem Aktionen zugeordnet sind. Befindet sich der Nutzer beispielsweise in der Startansicht des GUIs, so sind mögliche Eingaben „Öffne Task 1“, „Lösche Task 1“ oder „Wähle Roboter 1 aus“. Wird die Eingabe „Öffne Task 1“ getätigt, so wird im Hintergrund der Task aus der Datenbank abgerufen, als aktueller Task gekennzeichnet und die Ansicht wechselt in die Skill-Übersicht. Hier wird ein neues Eingabealphabet generiert, das beispielsweise die Eingaben „Gehe zur Startansicht“, „Neuen Skill hinzufügen“ oder „Lösche Skill Bohren“ beinhalten könnte.

Abbildung 3.9 zeigt die (vereinfachte) Dialoghierarchie. Die einzelnen Aktionen werden zur Anwendung im AR-Interaktionskonzept gruppiert, sodass beispielsweise in der Startansicht zunächst die Gruppe „Tasks“ angewählt werden muss, bevor die Aktionen „Neuer Task“ oder „Öffne MyTask“ zur Verfügung stehen.

Bei jeder Zustandsänderung wird so ein Objekt erzeugt, das den Zustand mit Namen, Beschreibung, Bild und Kennung enthält, sowie die Liste aller Eingaben, die wiederum mit Namen, Beschreibung, Bild und Kennung gespeichert werden. Dieses Objekt wird anschließend mittels des WebSocket-Protokolls an den eigentlichen IOService gesendet, bei dem es sich, wie in Abschnitt 3.1.1 beschrieben, um einen Webservice mit REST-Interface handelt.

Der IOService kommuniziert mittels *Links&Nodes* (siehe Abschnitt 3.1.1) mit einer beliebigen Anzahl an Eingabegeräten. Neben dem hier vorgestellten AR- Interaktionskonzept können beispielsweise ein Modul zur Spracheingabe oder auch ein simpler Fußschalter an den IOService angebunden werden. Dieser gibt das Zustandsobjekt an die Eingabegeräte weiter und wartet auf eine Antwort in Form der Kennung der ausgeführten Eingabe. Die Kennung wird anschließend an das Frontend weitergeleitet, das die entsprechenden Aktionen ausführt.

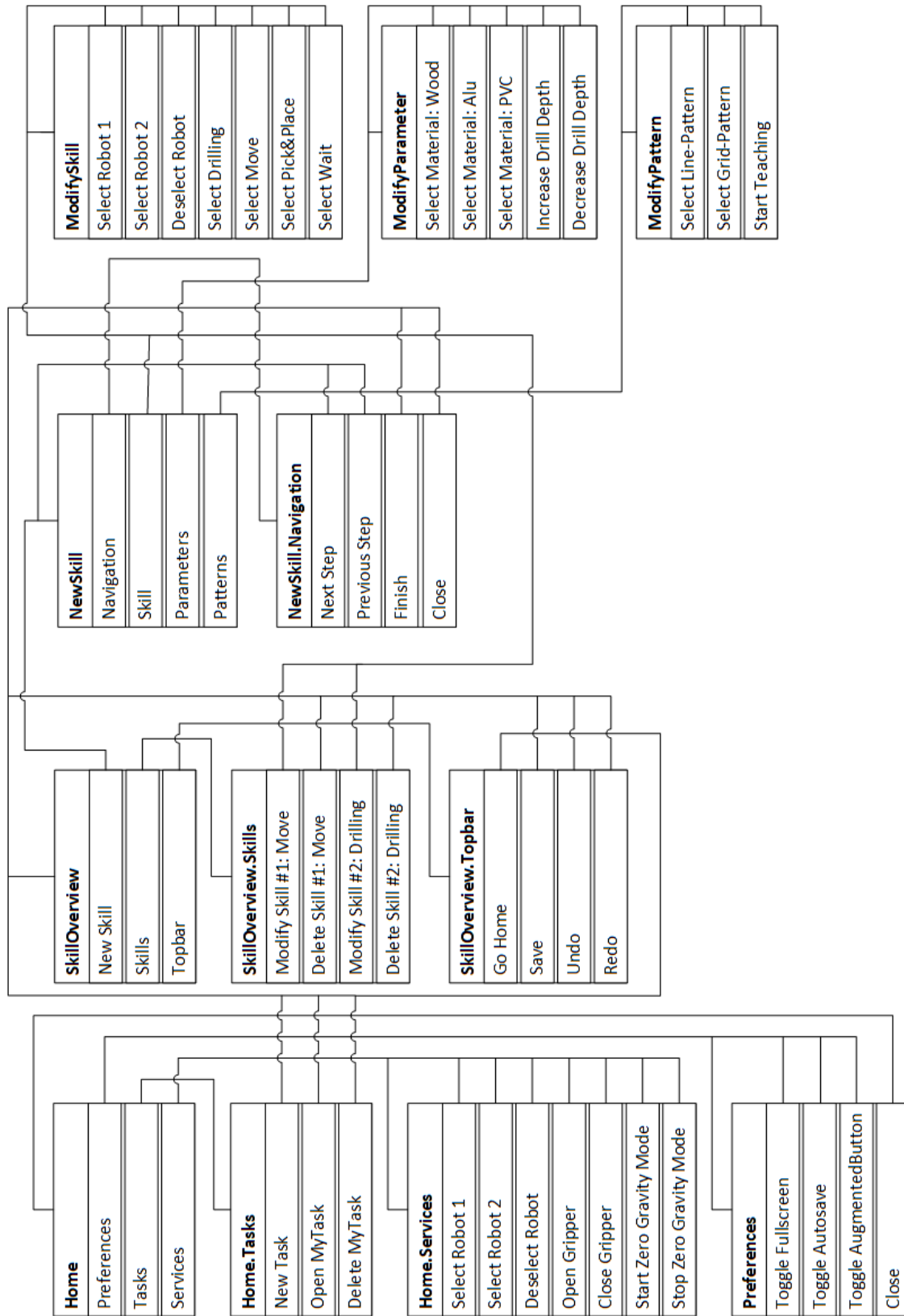


Abb. 3.9.: Dialoghierarchie RAZER

Augmented Button Service

Anforderungen

- Die verfügbaren Eingaben werden gruppiert und an die Datenbrille gesendet
- Die Daten des Roboters werden ausgelesen
 - Die Roboterpose wird an die Datenbrille übermittelt
 - Übersteigen die externen Kräfte einen bestimmten Schwellwert, wird der IOService informiert

Umsetzung Der ABService erhält vom IOService das Zustandsobjekt mit den verfügbaren Aktionen. Er wählt daraus Aktionen aus, die dem Nutzer zur Eingabe zur Verfügung stehen sollen. Aktionen werden dabei sinnvoll gruppiert. So kann der Nutzer beispielsweise beim Startbildschirm zwischen der Gruppe *Tasks* und *Services* wählen. Erst nach der Auswahl erhält er die Möglichkeit, auf weitere Aktionen zuzugreifen, so kann er beispielsweise einen bestimmten Task öffnen oder löschen. Da nur sechs Eingaberichtungen zur Verfügung stehen, wird dem Nutzer bei mehr als sechs Aktionen die Möglichkeit gegeben, die weiteren Aktionen über eine *Mehr*-Aktion anzuzeigen.

Die bis zu sechs Aktionen sendet der ABService mittels des AOA-Protokolls an die Datenbrille. Das AOA-Protokoll ermöglicht die Interaktion externer USB-Hardware (hier: der PC, auf dem RAZER läuft) mit Android-Geräten (hier: die Epson Moverio BT-200 Datenbrille). Das Android-Gerät wird anhand einer Vendor-ID und einer Product-ID identifiziert. Nachdem das Protokoll angefordert wurde, werden Strings mit Identifikatoren für Hersteller, Modellname, Beschreibung, Version, URL und Seriennummer der USB-Hardware übertragen. Sind diese so im Android-Manifest der Datenbrille hinterlegt, wird der Accessory-Modus gestartet. Android-Gerät und USB-Hardware können dann Daten miteinander austauschen. Der ABService wandelt das Eingabe-Objekt

dazu in einen JavaScript Object Notation (JSON)-String um und stellt eine Kennung voran, der zur Identifikation der Art der übertragenden Daten dient.

Sobald die Daten an die Datenbrille gesandt wurden, wird ein Thread gestartet, der mit einer vorgegebenen Rate die Roboterparameter ausliest. Diese werden mittels eines Interfaces ausgelesen, das *Links&Nodes*-Parameter verwendet. Die Roboterparameter sind zum einen die Gelenkkonfiguration zur Anpassung der virtuellen Inhalte und zum anderen die externen Kräfte, die auf den Roboter wirken. Die externen Kräfte sind in Bezug zum Endeffektor des Roboters angegeben. Da Bezug der Interaktion jedoch das Gelenk vor dem Endeffektor ist, müssen die Kräfte zunächst umgerechnet werden. Es gilt

$$F_C = R_\alpha^{-1} \cdot {}^0_7R^{-1} \cdot F_{ext} \quad (3.3)$$

mit F_C als Kraftvektor am Eingabegelenk, F_{ext} als externer Kraftvektor am Endeffektor, 0_7R als Rotationsmatrix zwischen Basis und Endeffektor und R_α als Rotationsmatrix der x-y-Ebene von der Basis zum Endeffektor. Diese kann mit

$$R_\alpha = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.4)$$

berechnet werden. α ist dabei durch

$$\alpha = \arctan \left(\frac{{}^1_6p_y}{{}^1_6p_x} \right) \quad (3.5)$$

gegeben.

Zur Umsetzung des in Abschnitt 3.2.2 beschriebenen basisorientierten Eingabekonzepts ist die Bestimmung der Transformationsmatrix 0_6T zwischen Basis und dem Gelenk vor dem Endeffektor vonnöten. Dies erfolgt mithilfe von Gleichung 3.1 und Gleichung 3.2.

Drückt der Nutzer gegen den Endeffektor, steigen die externen Kräfte an. Überschreiten sie für eine bestimmte Richtung einen Grenzwert, wird der IOService über die Eingabe informiert, indem die Kennung der zur betätigten Richtung zugeordneten Eingabe übertragen wird. Um erneut eine Eingabe tätigen zu können, muss der Grenzwert zeitweilig unterschritten werden.

Die Roboterkonfiguration wird kontinuierlich an die Datenbrille übertragen. Auch hier wird eine Kennung vorangestellt, um der Datenbrille die Identifikation der Daten zu erlauben.

Datenbrille

Anforderungen

- Die Pose von Markern in Weltkoordinaten im Sichtbereich wird bestimmt
- 3D-Modelldaten werden perspektivisch an die Pose des Roboters angepasst eingeblendet
- 3D-Modelldaten können aus .obj-Dateien geladen werden
- Die dargestellten Inhalte passen sich dem aktuellen Eingabealphabet an
- Die Darstellung beinhaltet Name und Bild der verfügbaren Eingaben
- Nicken und Kopfschütteln des Nutzers werden registriert

Tab. 3.1.: Vergleich der AR-Frameworks ARToolKit und Vuforia (+ positiv, o neutral, - negativ)

| | ARToolKit | Vuforia |
|------------------------|-----------|---------|
| Android Unterstützung | + | + |
| Datenbrillen | + | + |
| Dokumentation | o | + |
| Marker | - | + |
| Lizenz | + | o |
| Bedienerfreundlichkeit | o | + |

AR-Framework Zur Umsetzung der beschriebenen Anforderungen wird ein geeignetes AR-Framework ausgewählt. Entscheidend ist die Unterstützung von Android sowie die Möglichkeit der Verwendung mit einer Datenbrille. Es wurden lediglich frei verfügbare Frameworks in Betracht gezogen. Zur Auswahl standen ARToolKit und Vuforia. Tabelle 3.1 zeigt Vor- und Nachteile der beiden Frameworks auf. Während beide Android unterstützen und Funktionen zum Einsatz von Datenbrillen bereitstellen, zeichnet sich Vuforia insbesondere durch die Größe der Community und damit der Verfügbarkeit von Tutorials und Foren aus. ARToolKit erlaubt ausschließlich die Verwendung einfacher Marker, während Vuforia das Tracking von 3D-Objekten, Bildern, Text und VuMarks (ähnlich zu QR-Codes) ermöglicht. Nachteilig bei Vuforia ist die Lizenzierung. So kann das Framework zwar im Entwicklungsstadium kostenlos genutzt werden, für eine kommerzielle Verwendung muss jedoch eine Lizenz erworben werden. ARToolKit ist hingegen eine Open Source Software unter der LGPL-Lizenz ². Als letztes Kriterium wurde die subjektive Bedienerfreundlichkeit herangezogen. Dazu wurden beide Frameworks verwendet, um ein einfaches AR-Programm auf der Epson Moverio BT-200 in Betrieb zu nehmen. Dabei schloss Vuforia aufgrund der leichten Verständlichkeit und guten Übersicht etwas besser ab als das ARToolKit. Die Wahl fiel aufgrund der überwiegend positiven Bewertungen zu Gunsten von Vuforia aus.

²Library General Public License

Marker Es gilt, geeignete Marker zur Anzeige der virtuellen Daten zu erstellen. Die Marker müssen von der Größe so gewählt sein, dass sie gut am Roboter angebracht werden können und dennoch vom Algorithmus detektiert werden. Zudem müssen die Marker unterschieden werden können, um die virtuellen Inhalte je nach Sicht auf den Roboter anzupassen. Eine Herausforderung ist die geringe Auflösung der Kamera in der Datenbrille, die die Detektion der Marker erschwert. Vuforia ermöglicht die Detektion von einfachen 3D-Objekten, Bildern, Text und VuMarks. Da es sich bei dem Roboter um kein einfaches 3D-Objekt, wie beispielsweise ein Zylinder oder Quader handelt, fällt diese Option aus. Die Detektion von Bildern ist geeignet, um etwa Produktverpackungen zu detektieren. Zu einer sicheren Erkennung werden Grafiken mit hohem Detailreichtum benötigt. Da sie relativ groß sein müssen, sind sie für den Anwendungsfall in dieser Masterarbeit nicht geeignet. Auch Text würde zu großen Dimensionen und zu Ablenkung des Nutzers führen. Daher fiel die Wahl auf die Erstellung einer VuMark. VuMarks bestehen aus einer Kontur, Code-Elementen und einem Hintergrundbereich, der frei gestaltet werden kann. Anhand der Code-Elemente können verschiedene Marker erzeugt und unterschieden werden. Abbildung 3.10 zeigt das in dieser Arbeit verwendete VuMark. Als Kontur wurde das Logo des DLR gewählt.

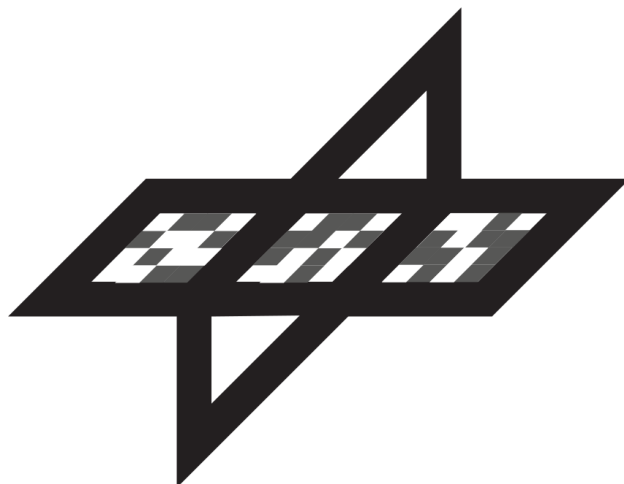


Abb. 3.10.: VuMark mit Code-Elementen

Umsetzung Beim Starten der Applikation auf der Datenbrille werden 3D-Objekte, Texturen und Marker geladen. Die 3D-Objekte wurden in einer 3D-Grafiksoftware modelliert und im Wavefront OBJ Dateiformat abgespeichert. Dieses speichert Eckpunkte, Normalen und Texturkoordinaten zeilenweise ab. Jeder Zeile wird eine Kennung zur Identifikation vorangestellt. Im Rahmen dieser Arbeit wurde eine Java-Klasse erstellt, die eine OBJ-Datei zeilenweise ausliest und die Daten in ein Format überführt, das von OpenGL genutzt werden kann, indem es Buffer mit den jeweiligen Eckpunkten, Normalen und Texturkoordinaten anlegt.

Beim Starten der Applikation wird ein Renderer erstellt. Dieser konfiguriert die Darstellungsfelder für das rechte und linke Display der Datenbrille und bindet Texturen und Shader zur Darstellung der virtuellen Inhalte. In einer Schleife wird zunächst die Anwesenheit von Markern im Sichtfeld der Kamera bestimmt und deren Pose an die Funktion übergeben, die die virtuellen Inhalte zeichnet. Abhängig von den aktuellen möglichen Interaktionen, der Roboterkonfiguration sowie der Position des detektierten Markers werden 3D-Richtungsobjekte, Icons und Text perspektivisch angepasst dargestellt.

Abbildung 3.11 zeigt das Mensch-Roboter-System mit den wichtigsten Koordinatensystemen. Zur Anzeige der virtuellen Inhalte ist die Transformation vom Bezugspunkt der Kamera zum Mittelpunkt des Robotergelenks K_GT relevant. Diese kann aus der Multiplikation der Transformation von Kamera zu Marker K_MT und von Marker zu Gelenk M_GT bestimmt werden.

$${}^K_GT = {}^K_MT \cdot {}^M_GT \quad (3.6)$$

Die Transformation K_MT ist durch das kamerabasierte Tracking bekannt. Die Transformation M_GT wird aus der Pose des Roboters sowie der Anbringung des Markers am Roboter bestimmt. Es gilt für die basisorientierte Interaktion

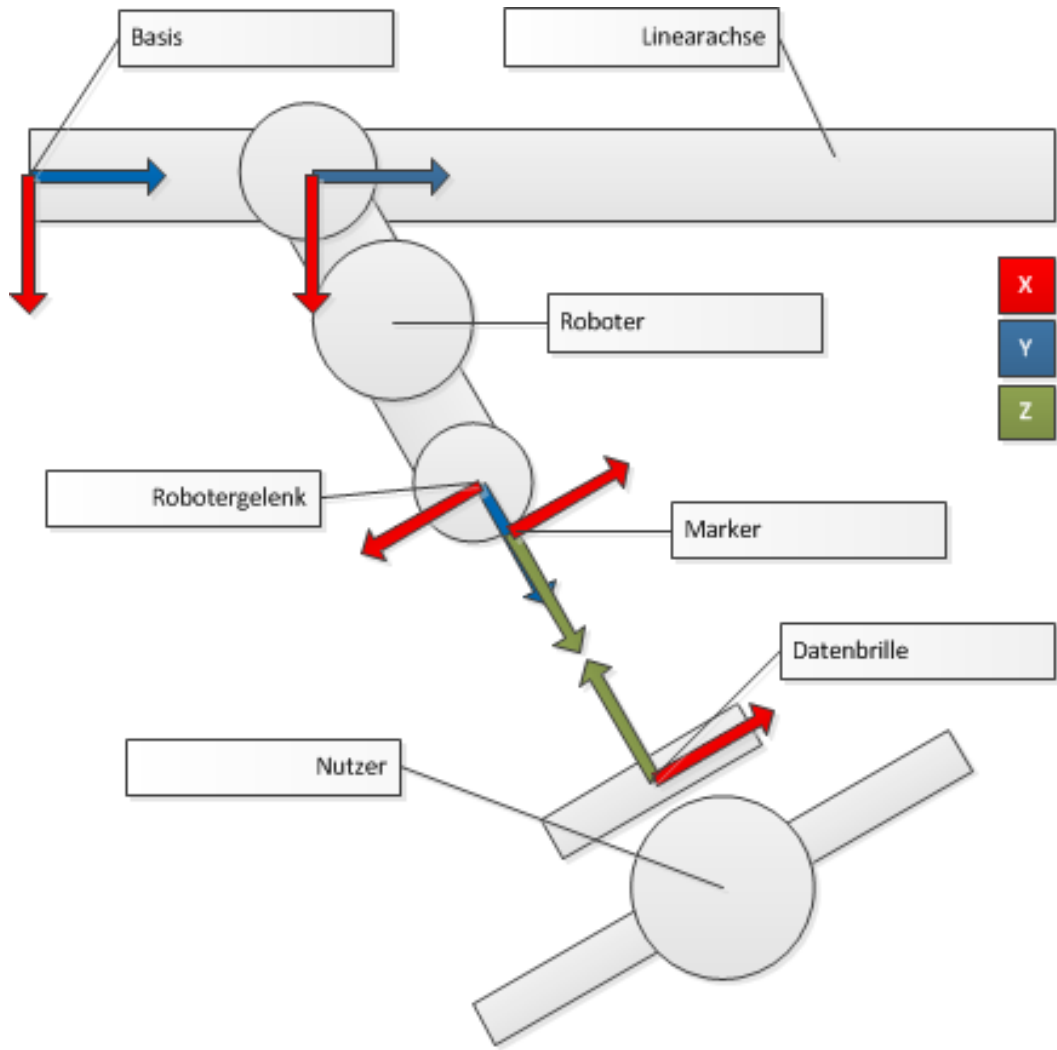


Abb. 3.11.: Darstellung der Koordinatensysteme am Mensch-Roboter-System

$${}^M_G T = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \left({}^0_7 R \cdot {}^6_7 R^{-1}\right)^{-1} \cdot R_\alpha & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -r_G \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.7)$$

Die Rotation von der Basis zum Endeffektor ${}^0_7 R$ kann unmittelbar aus den Parametern des Roboters ermittelt werden. Die Berechnung der Rotation vom Gelenk zum Endeffektor ${}^6_7 R$ ist durch Gleichung 3.1 gegeben. R_α beschreibt die Rotation von der Basis zum Endeffektor in der x-y-Ebene (siehe Gleichung 3.4). Die zweite Matrix in Gleichung 3.7 ist eine Konstante, die für

jeden Marker spezifisch ist. r_G bezeichnet dabei den Radius des Robotergetlenks, also den Abstand vom Mittelpunkt des Markers zum Mittelpunkt des Gelenks.

Die Aktualisierung des Eingabealphabets sowie der Roboterkonfiguration wird mithilfe eines Listener-Threads umgesetzt, der bei Nachrichten über das AOA-Protokoll die Texturen aktualisiert. Sie werden zur Darstellung von Text und Icons auf als Textur 2D-Flächen gelegt.

Abbildung 3.12 zeigt die Anzeige der Interaktionsoptionen am Roboter aus Sicht des Nutzers beispielhaft bei der Konfiguration eines neuen Skills. Das Bild wurde mithilfe eines VST-Geräts aufgenommen, spiegelt also nicht die reale Ansicht des Nutzers wider. Schwarze Elemente wie die Icons oder die Umrandung der Schrift sind auf einem OST-Gerät transparent, sind jedoch nötig, um Schrift bei der Überlagerung mit anderen virtuellen Elementen abzugrenzen. Die textuellen Informationen befinden sich stets im Vordergrund und werden durch ihre Umrandung von den anderen virtuellen Inhalten abgegrenzt. Sämtliche virtuelle Inhalte sind in weiß gezeichnet, sodass der für OST-Geräte maximale Kontrast zur realen Welt erzeugt wird. Text, Hand und Icon sind stets gruppiert, durch das geringe Sichtfeld der Datenbrille war es jedoch nötig, die einzelnen Gruppen dicht zusammenzurücken, sodass es teilweise zu Überdeckungen der Icons mit den Schriftzügen kommt.

In Abbildung 3.13 ist die Anzeige einer Fehlermeldung dargestellt, die vom Nutzer durch einen Druck nach rechts bestätigt werden kann. Wie oben erwähnt, wirken schwarze Elemente in der Ansicht in der Datenbrille transparent. Die Fehlermeldung ist nicht schwarz umrandet, da es zu keiner Überdeckung mit anderen virtuellen Elementen kommt und so der zusätzliche Rechenaufwand zur Erstellung der Umrandung gespart wird. Der weiße Rahmen stellt das Sichtfeld der Datenbrille dar. Um bei wichtigen Meldungen die

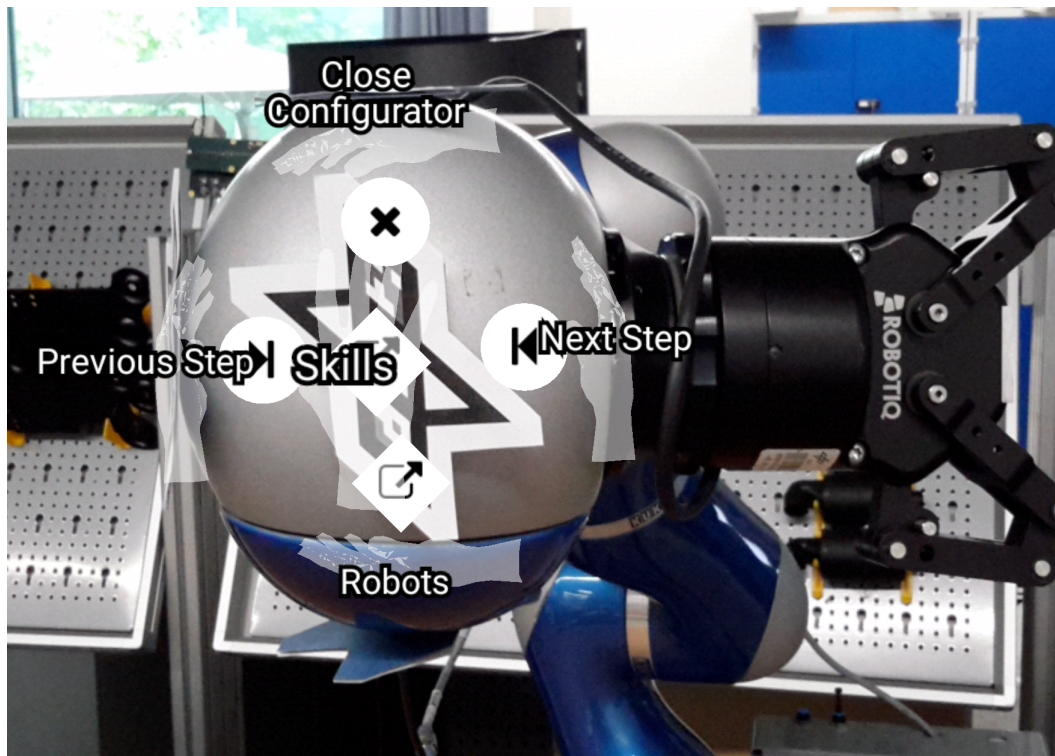


Abb. 3.12.: Anzeige der Interaktionsoptionen am Roboter

Aufmerksamkeit des Nutzers auf sich zu ziehen, werden die Fehlermeldungen groß im oberen Bereich des Sichtfelds dargestellt.

Befindet sich der Roboter im Gravitationskompensationsmodus, ist die Eingabe durch Drücken des Robotergelenks nicht mehr möglich, da dieses keinen Widerstand leistet. Um eine Pose des Roboters bestätigen zu können oder den Gravitationskompensationsmodus abubrechen, werden daher die Gesten Kopfnicken und Kopfschütteln verwendet, die durch die IMU der Datenbrille erfasst werden. Die nötige Software dazu wurde in einer Bachelorarbeit am DLR [45] entwickelt und für die Anwendung in dieser Arbeit modifiziert und eingebunden. Beim Wechsel des Roboters in den Gravitationskompensationsmodus erhält der Nutzer eine Einblendung über die AR-Datenbrille, die ihn über die aktuelle Aufgabe und die Möglichkeit der Bestätigung beziehungsweise des Abbruchs über die Gestensteuerung informiert.

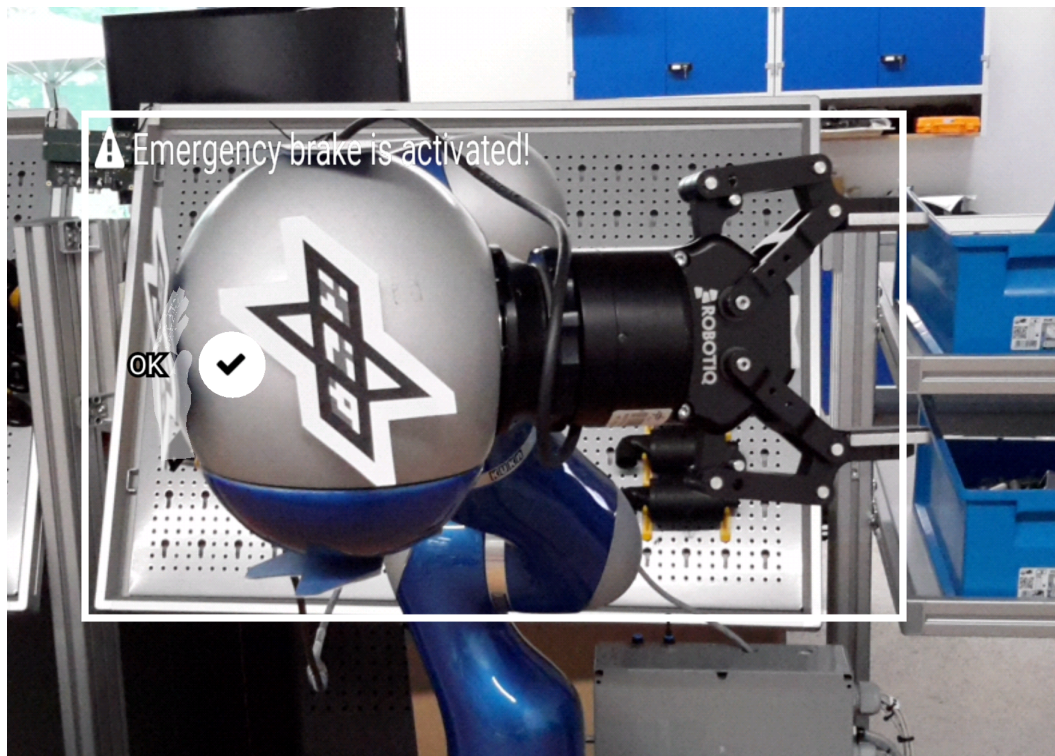


Abb. 3.13.: Anzeige einer Fehlermeldung am Roboter mit Sichtfeld der Datenbrille

Kommunikation

Abbildung 3.14 zeigt das Sequenzdiagramm der Initialisierung des Systems. Das Frontend verbindet sich zunächst mit dem IOService. Wird der *Augmented Button* in den Einstellungen des Frontends aktiviert, startet dieser den ABService und verbindet sich mit der Datenbrille. Diese startet bei der Initialisierung einen Thread, um mit dem ABService zu kommunizieren, wenn der Zustand des Frontends geändert wird. Der ABService startet wiederum einen Thread, der die Parameter des Roboters ausliest, um eine etwaige Eingabe zu erkennen.

Abbildung 3.15 zeigt das Sequenzdiagramm im Fall einer Eingabe durch einen Druck auf das Roboter gelenk. Innerhalb eines Threads im ABService werden die Roboterparameter ausgelesen. Übersteigen diese einen bestimmten Grenzwert, so informiert der ABService den IOService, der wiederum das Frontend informiert. Hier wird die ausgewählte Aktion ausgeführt, was

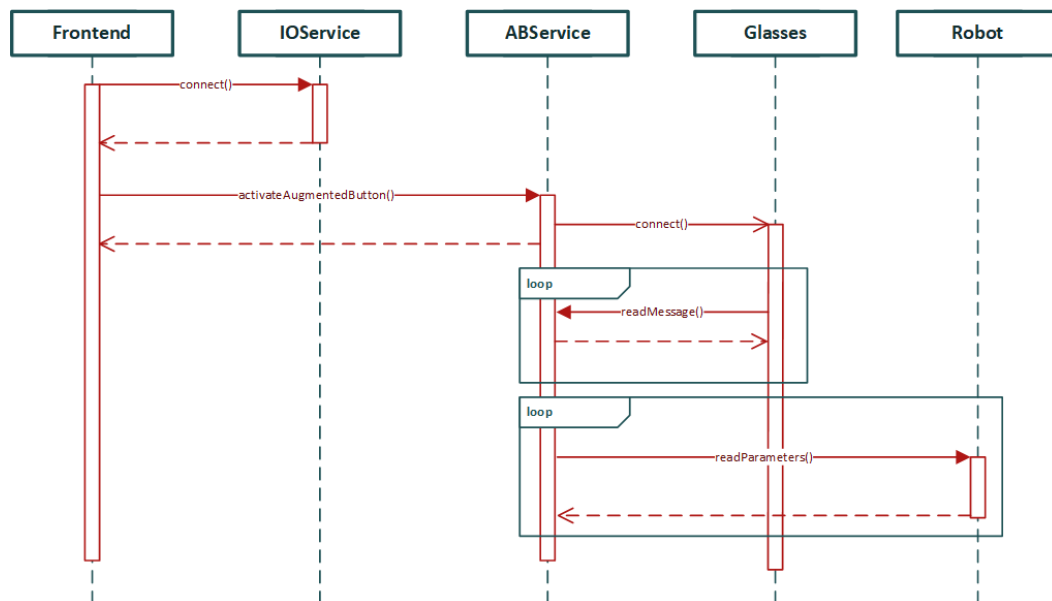


Abb. 3.14.: Sequenzdiagramm der Initialisierung

zu einer Änderung des Zustands führt. Das Frontend informiert daraufhin den IOService, der wiederum sämtliche angeschlossenen Eingabegeräte informiert, also auch den ABService. Dieser kommuniziert mit der Datenbrille, sodass die Anzeige an die aktuellen Aktionen angepasst wird.

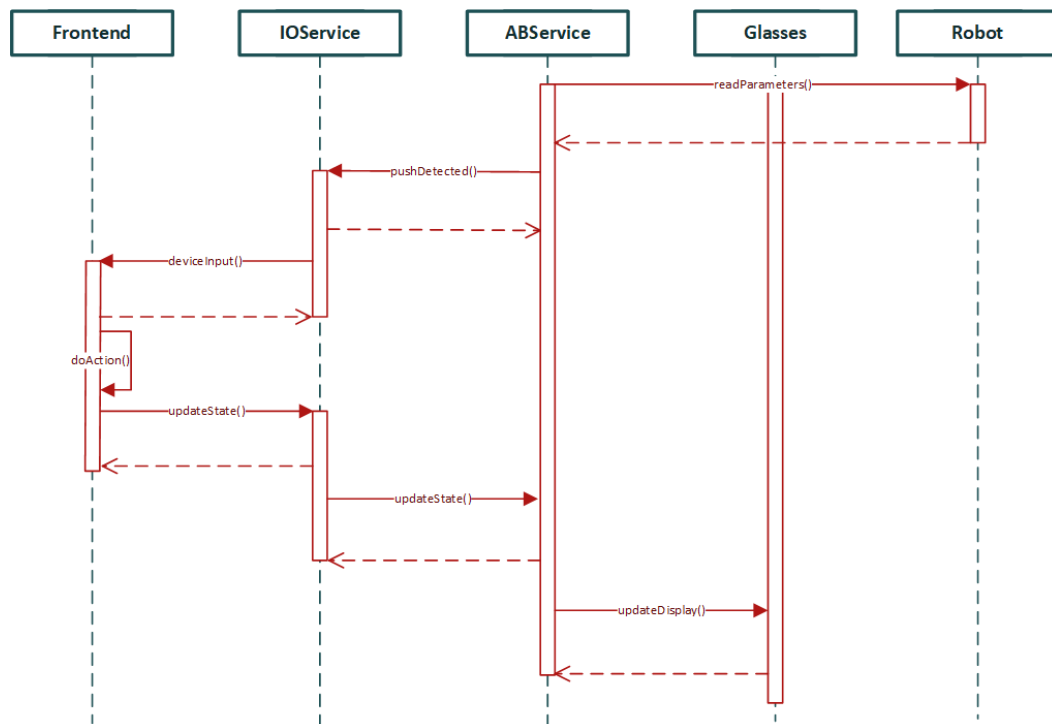


Abb. 3.15.: Sequenzdiagramm bei Eingabe

3.3 Versuchsdesign

In einer Nutzerstudie soll die Gebrauchstauglichkeit des AR-Interaktionskonzepts im Vergleich zum herkömmlichen System erfasst werden. Beim herkömmlichen System wird zur Programmierung eine grafische Oberfläche auf einem Tabletcomputer sowie PdV verwendet. Das AR-Interaktionskonzept erweitert das herkömmliche System um die Möglichkeit, Eingaben unmittelbar am Roboter zu tätigen.

Die Gebrauchstauglichkeit wird anhand subjektiver und objektiver Daten erhoben. Für die subjektive Einschätzung der Gebrauchstauglichkeit des Systems wird ein standardisierter Fragebogen (QUEAD [47], siehe Abschnitt A.2) eingesetzt. Dieser umfasst 26 Aussagen, in denen der wahrgenommene Nutzen, die wahrgenommene Bedienbarkeit, die Emotionen, die Einstellung und der Komfort bewertet werden. Die Effizienz wird mittels der Bearbei-

tungszeit einer Testaufgabe gemessen. Zusätzlich werden zur Beurteilung der Effektivität die Bedienfehler während des Versuchs aufgenommen.

Abbildung 3.16 zeigt den Versuchsablauf der Nutzerstudie. Zunächst werden demographische Daten sowie Erfahrungen der Nutzer hinsichtlich des Umgangs mit Touchscreens und AR-Systemen erfasst (siehe Abschnitt A.2). Die AR-Datenbrille wird bei jedem Versuchsteilnehmer kalibriert, um Effekte durch unterschiedliche Gesichtsgeometrien zu vermeiden. Während einer Einführung können sich die Nutzer im Umgang mit den Systemen vertraut machen, indem eine definierte Aufgabe mit einer Schritt-für-Schritt-Anleitung bearbeitet wird.

In einer zweiten Masterarbeit parallel zu dieser wurde ein weiteres Interaktionskonzept entwickelt. Dieses erweitert das herkömmliche System um die Möglichkeit der Spracheingabe. Alle Probanden erprobten zunächst das herkömmliche System mit der grafischen Oberfläche auf dem Tabletcomputer. Anschließend nutzten drei der Probanden das in dieser Arbeit entwickelte Interaktionskonzept und vier der Probanden die Spracheingabe, bevor sie abschließend das noch nicht erprobte System nutzten.

Zur Erprobung wurde eine Testaufgabe durchgeführt, in der der Proband eine Sequenz bestehend aus zwei Skills programmiert.

Skill 1 Bewegung von Roboter 1 in eine definierte Pose mit einer Geschwindigkeit von 0.6 [m/s]

Skill 2 Pick&Place eines Objekts von Position A zu Position B mit Roboter 1

Skill 1 beinhaltet folgende Teilaufgaben:

- Öffnen des Skillkonfigurators

- Auswahl des Roboters (Roboter 1) und des Skills (Moving)
- Parametrisierung des Skills (Geschwindigkeit = 0.6 [m/s])
- Vormachen der Pose
- Speichern des Skills

Skill 2 beinhaltet folgende Teilaufgaben:

- Öffnen des Skillkonfigurators
- Auswahl des Roboters (Roboter 1) und des Skills (Pick&Place)
- Vormachen der Pick-Pose
- Vormachen der Place-Pose
- Wegbewegen des Roboterarms vom Objekt
- Speichern des Skills

Bei der Verwendung des AR-Interaktionskonzepts sowie des Systems mit der Möglichkeit der Spracheingabe wurden die Nutzer aufgefordert, die grafische Oberfläche auf dem Tabletcomputer so wenig wie möglich zu verwenden. Nach dem Abschließen der Testaufgaben beantworteten die Probanden die Fragebögen zur Gebrauchstauglichkeit.

Unabhängige Variable des Versuchs ist das Interaktionskonzept und abhängige Variablen die Bewertung der Gebrauchstauglichkeit sowie die Bearbeitungszeit der Programmieraufgaben.

An der Studie nahmen sieben Probanden teil. Es handelt sich um Studenten, die zur Zeit der Studie am DLR tätig waren. Fünf Probanden sind männlich, zwei weiblich. Das Alter der Probanden beträgt 24.1 ± 1.9 Jahre. Alle sind rechtshändig und benötigen eine Brille zur Korrektur der Sehstärke. Zwei haben in der Vergangenheit noch nie ein AR-Gerät verwendet, alle anderen selten. Alle Teilnehmer nutzen regelmäßig Touchscreens. Da es sich bei den Probanden um Studenten des Zentrums für Robotik und Mechatronik am

DLR handelt, ist davon auszugehen, dass die Probanden neuen Technologien nicht abgeneigt sind und eine Affinität zu Robotern besitzen, jedoch keine konkrete Erfahrung in der Roboterprogrammierung haben.

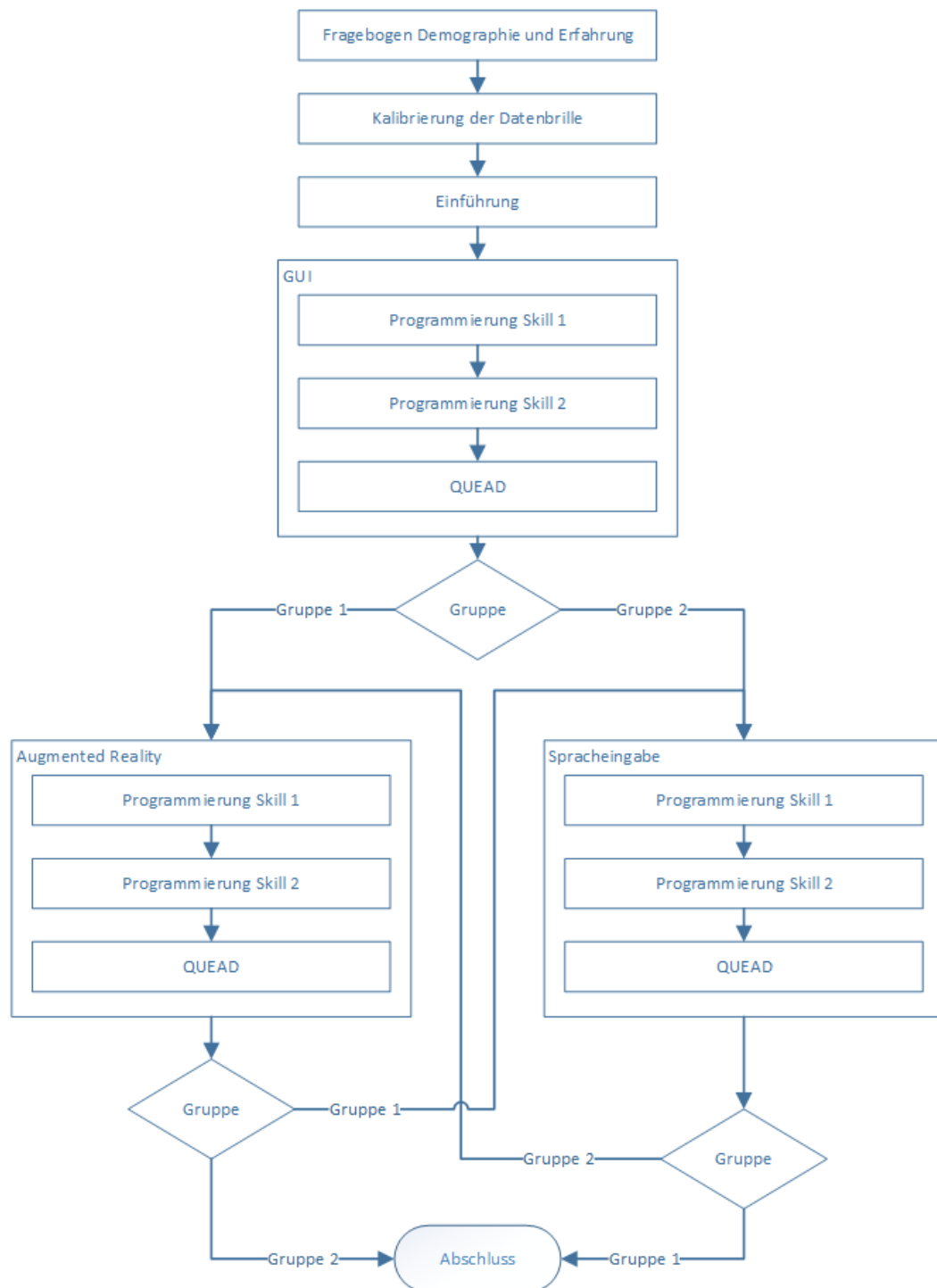


Abb. 3.16.: Versuchsablauf der Nutzerstudie

Ergebnisteil

Aufgrund des geringen Stichprobenumfangs und der erwarteten Störeinflüssen durch Erprobung des dritten Systems sowie der Versuchslänge von etwa 90 Minuten pro Proband werden die Ergebnisse deskriptiv dargestellt.

4.1 Subjektive Daten

In Abbildung 4.1 und Tabelle 4.1 sind die Ergebnisse der Fragebögen zur Gebrauchstauglichkeit nach Kategorien aufgeteilt dargestellt. Dazu wurden die Bewertungen negativ formulierter Aussagen angepasst, sodass für alle Fragen 1 die beste und 5 die schlechteste Bewertung ist. Die Bewertungen der einzelnen Aussagen sind in Abschnitt A.1 in den Abbildungen A.1 und A.2 dargestellt. Der Mittelwert der Bewertungen des herkömmlichen Systems ist in allen Kategorien höher als der Mittelwert der Bewertungen des AR-Interaktionskonzepts. Letztere weisen durchgehend eine hohe Spannweite auf. Auffallend gut wurden für das AR-Interaktionskonzept die Aussagen „I was able to perform precise motions with this system“, „It was easy to learn to use this system“ und „I think that using this system is a good idea“ bewertet.

Tab. 4.1.: Mittelwerte und Standardabweichungen der Bewertungen der Gebrauchstauglichkeit von 1 (schlecht) bis 5 (sehr gut)

| | GUI | AR |
|---------------|-----------------|-----------------|
| Nutzen | 4.33 ± 0.50 | 3.10 ± 0.91 |
| Bedienbarkeit | 4.41 ± 0.22 | 3.27 ± 0.92 |
| Emotionen | 4.71 ± 0.28 | 4.05 ± 0.73 |
| Einstellung | 4.57 ± 0.32 | 3.76 ± 1.07 |
| Komfort | 4.76 ± 0.50 | 4.19 ± 0.88 |

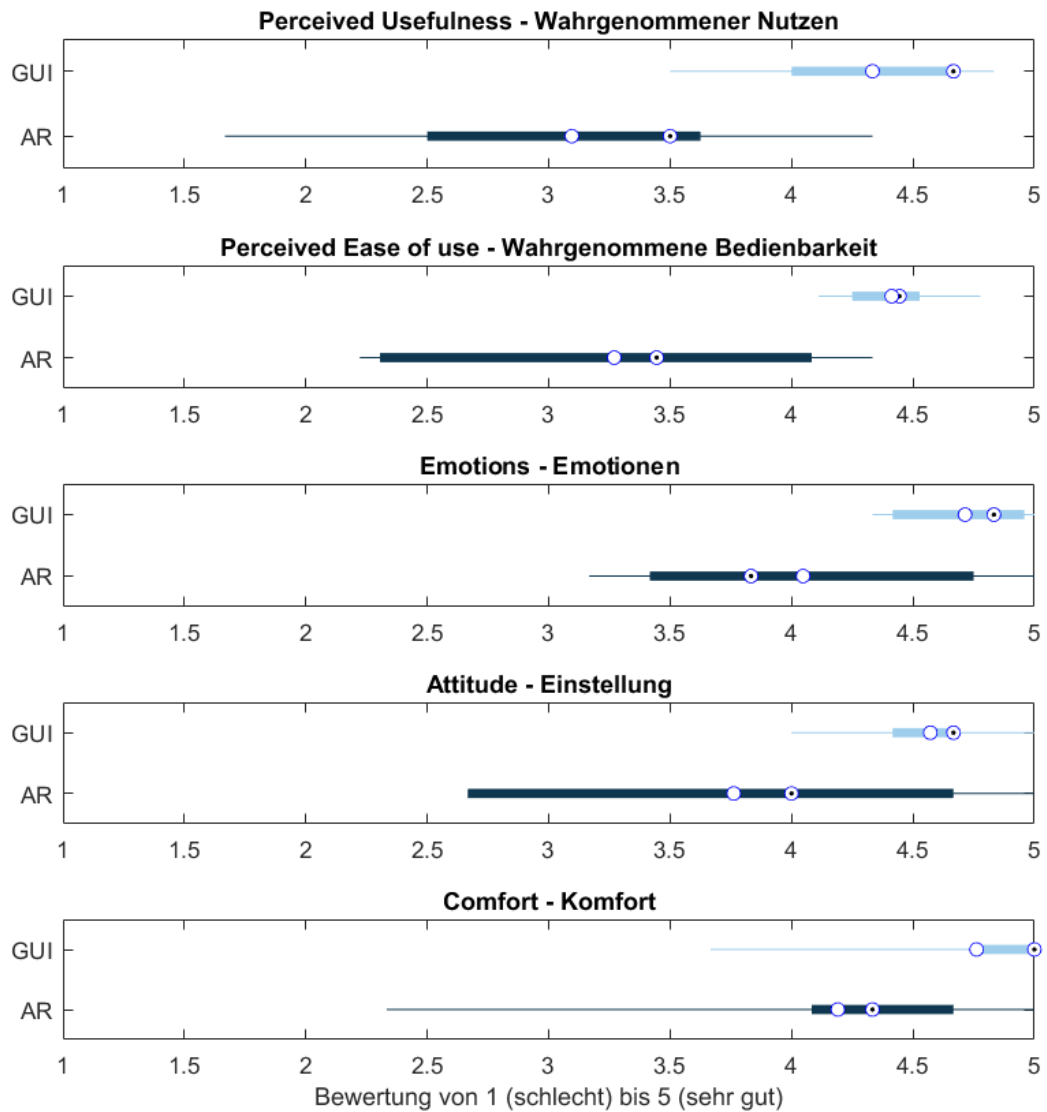


Abb. 4.1.: Ergebnisse der Bewertung der Gebrauchstauglichkeit mit Mittelwert (○), Median (⊙), unterem und oberem Quartil (■) und Spannweite (—)

4.2 Objektive Daten

4.2.1 Bearbeitungszeiten

In Abbildung 4.2 und Tabelle 4.2 sind die Bearbeitungszeiten der Testaufgabe dargestellt. Für die Erledigung der gesamten Testaufgabe mit dem AR-Interaktionskonzept benötigen die Probanden im Schnitt mehr als doppelt so lange wie zur Programmierung mit der GUI auf dem Tabletcomputer. Wird

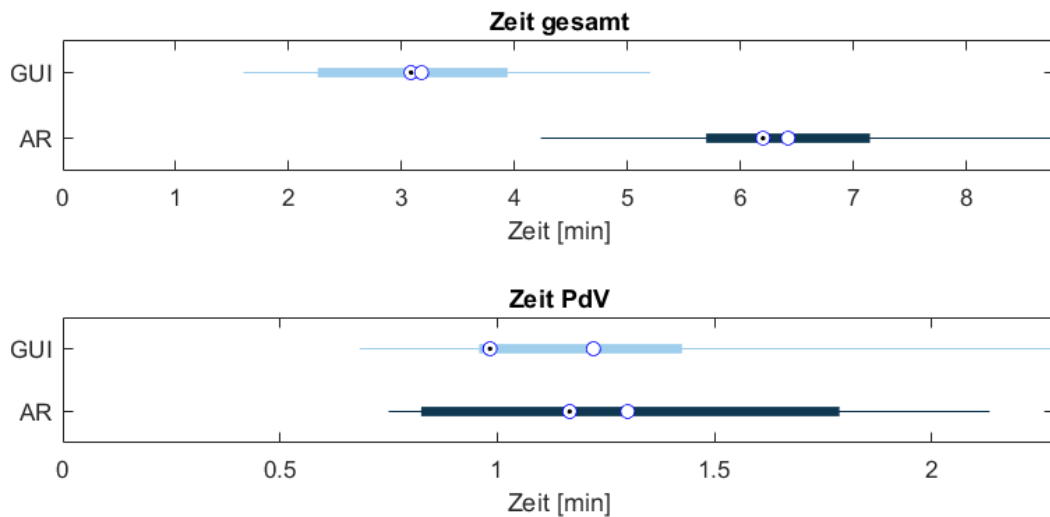


Abb. 4.2.: Bearbeitungszeiten mit Mittelwert (○), Median (⊙), unterem und oberem Quartil (■) und Spannweite (—)

Tab. 4.2.: Mittelwerte und Standardabweichungen der Bearbeitungszeiten in Minuten

| | GUI | AR |
|-------------------|-----------------|-----------------|
| Zeit gesamt [min] | 3.18 ± 1.24 | 6.42 ± 1.42 |
| Zeit PdV [min] | 1.22 ± 0.53 | 1.30 ± 0.55 |

jedoch die Bearbeitungszeit des PdV betrachtet, weichen die Mittelwerte beider Systeme nur noch geringfügig voneinander ab. Vier der sieben Probanden waren beim PdV mit dem AR-Interaktionskonzept im Schnitt sogar schneller.

4.2.2 Bedienfehler

Während des Versuchs zeigten sich folgende Bedienfehler bei der Verwendung des AR-Interaktionskonzepts:

- Zwei der Probanden wandten nicht genug Kraft zur Betätigung des Roboters auf
- Nach der Beendigung des Gravitationskompensationsmodus versuchten drei Probanden einen Dialog mit einem Kopfnicken zu bestätigen

- Zwei der Probanden versuchten im Gravitationskompensationsmodus eine Eingabe durch Drücken des Roboters durchzuführen
- Ein Proband bewegte während des Gravitationskompensationsmodus den Kopf so stark, dass ein unfreiwilliges Nicken detektiert wurde

Diskussion und Zusammenfassung

5.1 Diskussion der Ergebnisse

5.1.1 Versuchsdesign

Durch die recht lange Dauer (etwa 90 Minuten pro Proband) ist mit Ermüdungserscheinungen zum Ende des Versuchs zu rechnen. Jedoch wird in [46] gezeigt, dass die Bewertungen von Nutzern nach 90-minütigen Sitzungen noch zuverlässig waren, obwohl die Probanden schon deutliche Anzeichen von Ermüdung zeigten.

Auch wenn der Stichprobenumfang zur Anwendung schließender Statistik zu gering ist, zeigten Nielsen und Landauer [36], dass sich mit einer Anzahl von sieben Probanden bereits 93% der Gebrauchstauglichkeits-Probleme aufdecken lassen. Diese können in einer Weiterentwicklung des Systems adressiert werden.

5.1.2 Objektive Daten

Effizienz

Die Effizienz des AR-Interaktionskonzepts wurde anhand der Bearbeitungszeit ermittelt. Die durchschnittliche Bearbeitungszeit für die Testaufgabe war bei Verwendung des AR-Interaktionskonzept mehr als doppelt so groß wie

bei Verwendung der GUI auf dem Tabletcomputer. Die Probanden waren aufgefordert, bei der Bearbeitung der Testaufgabe mit dem AR-Interaktionskonzept die grafische Oberfläche auf dem Tabletcomputer so wenig wie möglich zu verwenden. So wurde auch die Parametrisierung verschiedener Skills, bei der der Nutzer eine große Anzahl an Eingabeoptionen hat, mit dem AR-Interaktionskonzept durchgeführt. Da dieses dem Nutzer maximal fünf Interaktionsoptionen auf einmal zur Verfügung stellen kann, benötigten die Probanden viel Zeit, um die Parametrisierungen vorzunehmen. Hier sind die bewährten Interaktionsparadigmen grafischer Oberflächen mit Schieberegler, Textfeldern und Dropdown-Menüs dem AR-Interaktionskonzept klar überlegen.

Betrachtet man die Bearbeitungszeiten beim PdV, bei dem der Nutzer im herkömmlichen System zwischen Bedienung des Roboters und des Tabletcomputers wechseln musste, lag der Mittelwert der Bearbeitungszeiten für das AR-Interaktionskonzept nur geringfügig über dem Mittelwert der grafischen Oberfläche. Hier kommen die Vorteile des AR-Interaktionskonzepts in Form der direkten Interaktion am Roboter und der ständig präsenten Anzeige der Informationen am stärksten zum Tragen.

Effektivität

Während der Durchführung der Testaufgaben mit dem AR-Interaktionskonzept traten verschiedene Bedienfehler auf (siehe Abschnitt 4.2.2). In der Regel erkannten die Nutzer nach kurzer Zeit, dass ihr Vorgehen sie nicht zum Ziel führt. Sie konnten die Aufgaben dennoch erfolgreich abschließen, sodass die Effektivität des Systems nicht beeinflusst wurde. Lediglich beim falsch registrierten Kopfnicken war eine Wiederholung des PdV vonnöten.

5.1.3 Subjektive Daten

Wahrgenommener Nutzen

Der wahrgenommene Nutzen des AR-Interaktionskonzepts wurde von den Probanden im Vergleich zur GUI auf dem Tabletcomputer deutlich schlechter bewertet. Dies ist auf die geringe Effizienz des Systems bei Aufgaben mit vielen Eingabeoptionen zurückzuführen, die aus den hohen Bearbeitungszeiten (siehe Abschnitt 4.2.1) hervorgeht. Die Beobachtungen decken sich mit den Ergebnissen von Wille et al. [61], die die Ausführung einer Montageaufgabe bei Verwendung eines Tabletcomputers und eines Head-Mounted-Displays verglichen. Auch hier war die Leistung bei Verwendung des Tabletcomputers besser.

Wahrgenommene Bedienbarkeit

Die wahrgenommene Bedienbarkeit wurde für das AR-Interaktionskonzept deutlich schlechter bewertet als für die GUI auf dem Tabletcomputer.

Das kleine Sichtfeld der Datenbrille erfordert die Darstellung der virtuellen Inhalte auf kleinstem Raum. Dies führte in der Umsetzung zu Überdeckungen der eingeblendeten Icons mit den beschreibenden Textelementen (siehe Abbildung 3.12), die zu einer schlechteren Bewertung der wahrgenommenen Bedienbarkeit geführt haben könnten.

Zwei der Probanden berichteten zudem von Doppelbildern, die zu einer Überladung der virtuellen Inhalte führten, sodass die Eingabemöglichkeiten nur noch schwer erkennbar waren. Wie in Abschnitt 2.3.7 beschrieben, ist das Auftreten von Doppelbildern auf eine mangelhafte Kalibrierung zurückzuführen. Zu Beginn des Versuchs wurde die Datenbrille für jeden einzelnen Probanden kalibriert. Dazu musste der Nutzer ein rotes virtuelles Rechteck

mit einem rechteckigen Kalibrierbild für jedes einzelne Auge und aus verschiedenen Distanzen in Deckung bringen. Am Ende des Kalibriervorgangs hat der Nutzer die Möglichkeit, die Kalibrierung anhand eines eingeblendeten virtuellen Objektes zu überprüfen. Möglicherweise sind die Probanden, die von Doppelbildern berichteten, nicht sorgfältig genug bei der Kalibrierung vorgegangen oder konnten die mangelhafte Kalibrierung bei der Überprüfung nicht erkennen.

Alle Probanden nutzen regelmäßig Touchscreens und sind daher mit den Interaktionsparadigmen grafischer Oberflächen vertraut. AR-Geräte wurden jedoch nie oder selten von den Teilnehmern genutzt. Obwohl sie vor dem Beginn der Studie in der Einleitung bereits einfache Aufgaben mit dem AR-Interaktionskonzept durchgeführt haben, müssen sich die Probanden während des Versuchs weiterhin an die Technologie gewöhnen. Wille et al. [61] berichten, dass bei einem Versuch mit einem Head-Mounted-Display nach zwei Sitzungen mit einer Länge von vier Stunden kaum Gewöhnungseffekte auftraten. Es ist daher davon auszugehen, dass eine deutlich längere Zeit zur Gewöhnung an derart neue Technologien vonnöten ist, damit sich eine Verbesserung in der Wahrnehmung der Bedienbarkeit zeigt.

Emotionen

Auch in der Kategorie Emotionen wurde das AR-Interaktionskonzept deutlich schlechter bewertet als die GUI auf dem Tabletcomputer. Ausschlaggebend waren hier vor allem die Bewertungen der Aussagen „I like using this system“ und „I feel comfortable using this system“. Thüring und Mahlke [57] zeigen, dass gut konzipierte Systeme sowie ein geringerer Grad an Neuheit positivere Gefühle hervorrufen. Die Bewertung der Emotionen könnte so durch die regelmäßige Verwendung von Touchdisplays und grafischen Oberflächen im Gegensatz zu AR-Geräten beeinflusst sein.

Einstellung

Auch die Bewertung der Einstellung ist für das AR-Interaktionskonzept deutlich weniger positiv als für die GUI auf dem Tabletcomputer. Vier der sieben Probanden lehnten eine Verwendung des Interaktionskonzepts bei zukünftigen Aufgaben ab. Da die Frage, ob die Nutzung des Systems B eine gute Idee sei, mit einer Wertung von durchschnittlich 4.1 Punkten auf einer Skala von 1 (schlecht) bis 5 (sehr gut) Zustimmung erfuhr, ist zu vermuten, dass die Ablehnung zukünftiger Verwendung nicht auf die Konzeptionierung an sich, sondern auf die aktuelle Umsetzung zurückzuführen ist. Verbesserungen der Umsetzung werden in Abschnitt 5.2 erläutert.

Komfort

Der Komfort des AR-Interaktionskonzepts wurde mit durchschnittlich 4.2 Punkten von allen Kategorien am besten bewertet. Der Wert liegt dennoch unter dem Wert des herkömmlichen Systems mit 4.8 Punkten. Alle Probanden in der Nutzerstudie waren Brillenträger, sodass sie die Datenbrille und ihre Brille zur Korrektur der Sehstärke gleichzeitig tragen mussten. Einer der Probanden berichtete, dass ihm das Tragen der Datenbrille Schmerzen bereite. Wille et al. [60] beobachteten, dass die Verwendung von Head-Mounted-Displays bei längerer Verwendung zu Kopf- und Nackenschmerzen führten, die Verwendung einer leichteren und kompakteren Datenbrille für denselben Einsatz jedoch nicht. Es ist daher davon auszugehen, dass durch die technologische Entwicklung der Trend zu leichteren und kleineren Datenbrillen fortgesetzt werden kann und sich negative Einflüsse auf die Gebrauchstauglichkeit durch schlechten Tragekomfort verringern.

5.2 Ausblick

Die Nutzerstudie hat verschiedene Gebrauchstauglichkeits-Probleme aufgedeckt, deren Behebung zur Ausreifung des Interaktionskonzepts notwendig sind.

Zur Vermeidung der Bedienfehler (siehe Abschnitt 4.2.2) ist unter anderem eine klarere Abtrennung der Bedienung im Gravitationskompensationsmodus nötig, sodass sich der Nutzer stets darüber bewusst ist, wann er die Gesten Kopfnicken und Kopfschütteln einsetzen kann und wann nicht. Dies kann beispielsweise durch eine gut sichtbare Einblendung erreicht werden. Alternativ wäre es auch möglich, die beiden Gesten außerhalb des Gravitationskompensationsmodus für Dialoge zuzulassen. Dazu müsste jedoch die Robustheit des Algorithmus zur Erkennung der Gesten gesteigert werden, damit die sie fälschlicherweise erkannt werden, wenn der Nutzer starke Kopfbewegungen durchführt.

Bei der Erkennung von Nutzereingaben über die externen Kräfte, die auf den Roboter wirken, wird bisher mit festen Schwellwerten gearbeitet. Zwei der Probanden drückten zu Anfang jedoch nicht fest genug gegen den Roboter, um diese Schwellwerte zu überschreiten. Hier wäre beispielsweise eine individuelle Anpassung der Sensitivität an die Präferenzen des Nutzers denkbar.

Wie in Abschnitt 3.2.3 erläutert ist ein Zusammenrücken der virtuellen Inhalte vonnöten, damit sie trotz des kleinen Sichtfelds der Datenbrille sichtbar sind. Dies führt teilweise zu Überdeckungen der Icons mit Schriftzügen. Die Implementierung eines Algorithmus, der übereinander liegende Elemente detektiert und im 3D-Raum verschiebt, könnte hier zur Sichtbarkeit aller Inhalte führen.

In Abschnitt 5.1.3 wurde beschrieben, dass bei zwei Probanden Doppelbilder auftraten, die auf eine fehlerhafte Kalibrierung zurückzuführen sind. Ob eine Kalibrierung erfolgreich durchgeführt wurde, kann nur durch den Nutzer selbst beurteilt werden, da sie von der individuellen Augengeometrie abhängig ist. Der Dialog zur Kalibrierung der Brille fordert den Nutzer am Ende auf, die Kalibrierung zu testen und gegebenenfalls zu wiederholen. Offensichtlich ist dieser Test jedoch nicht ausreichend, damit ein Erstbenutzer sicher eine unzureichende Kalibrierung erkennt. Der Kalibriervorgang müsste daher so gestaltet werden, dass der Nutzer unmissverständlich erkennt, ob die Kalibrierung erfolgreich durchgeführt wurde oder nicht.

Die hohen Bearbeitungszeiten und die Bewertung der Gebrauchstauglichkeit mittels Fragebögen deuten darauf hin, dass das AR-Interaktionskonzept für Aufgaben mit vielen Eingabemöglichkeiten, beispielsweise der Parametrisierung von Skills, der GUI auf dem Tabletcomputer unterlegen ist. Eine Weiterentwicklung des Interaktionskonzepts sollte daher den Fokus auf Programmierpassagen legen, bei denen nur wenige Eingabemöglichkeiten vorhanden sind und sich der Nutzer bereits in Interaktion mit dem Roboter befindet. Hier zeigte sich bei den Bearbeitungszeiten und den Bewertungen der Nutzer das größte Potential.

Das unter den oben genannten Gesichtspunkten weiterentwickelte System könnte dann in einer weiteren Nutzerstudie untersucht werden. Ein größerer Stichprobenumfang ließe dabei die Anwendung schließender Statistik zu. Um Effekte durch die Neuartigkeit des AR-Interaktionskonzepts zu vermeiden, sollte eine längere Trainingsphase zu Beginn der Studie durchgeführt werden. Zudem wäre eine repräsentativere Auswahl der Probanden wünschenswert.

Die AR-Technologie steckt noch weitestgehend in den Kinderschuhen und ist bisher nicht im breiten Markt angekommen. Es ist mit einer raschen Weiterentwicklung der Technologie – insbesondere in Hinsicht auf das Sichtfeld, die

Größe und das Gewicht der Brille sowie auf die Rechenleistung und die Effizienz der Algorithmen – zu rechnen, sodass die Umsetzung eines derartigen Konzepts vereinfacht wird.

5.3 Zusammenfassung

In dieser Arbeit wurde ein neuartiges Interaktionskonzept für die intuitive Roboterprogrammierung mittels AR und haptischer Eingabe entwickelt. Basierend auf einem integrierten Framework zur aufgabenorientierten Programmierung wird das Interaktionskonzept als eine Alternative zur Eingabe mittels eines Touchdisplays eingesetzt. Eingaben können durch das Interagieren mit dem Endeffektor des zu programmierenden Roboters getätigt werden. Einblendungen kontextabhängiger Informationen über eine AR-Datenbrille direkt am Endeffektor des Roboters leiten den Nutzer durch den Programmiervorgang. Dazu wurde ein Konzept zur Definition der Eingaberichtungen entwickelt und umgesetzt. Die möglichen Interaktionen werden dem Nutzer anhand dreidimensionaler Anzeigeelemente verdeutlicht, die er direkt in seinem Sichtfeld perspektivisch an die Umgebung angepasst angezeigt bekommt.

In einer Nutzerstudie wurde das Interaktionskonzept erprobt. Dabei ergaben sich deutlich schlechtere Bewertungen des Nutzens, der Bedienbarkeit und der Emotionen gegenüber der Programmierung mit einem GUI auf einem Tabletcomputer, wenn das AR-Interaktionskonzept als Ersatz für das herkömmliche System eingesetzt wird. Auch die Bearbeitungszeit zweier Testaufgaben war unter Verwendung des Interaktionskonzept doppelt so groß. Bei den Bearbeitungszeiten von Teilaufgaben mittels PdV lagen die Mittelwerte beider Systeme jedoch nur unwesentlich auseinander. Hier kommen die Vorteile des AR-Interaktionskonzepts in Form der direkten Interaktion am Roboter und der ständigen Verfügbarkeit der angezeigten Informationen

zum Tragen. Die Ergebnisse lassen darauf schließen, dass das AR-Interaktionskonzept in Situationen, in denen der Nutzer im direkten Kontakt mit dem Roboter ist, eine nützliche Ergänzung zum Touchdisplay darstellt.

Anhang

A.1 Ergebnisse

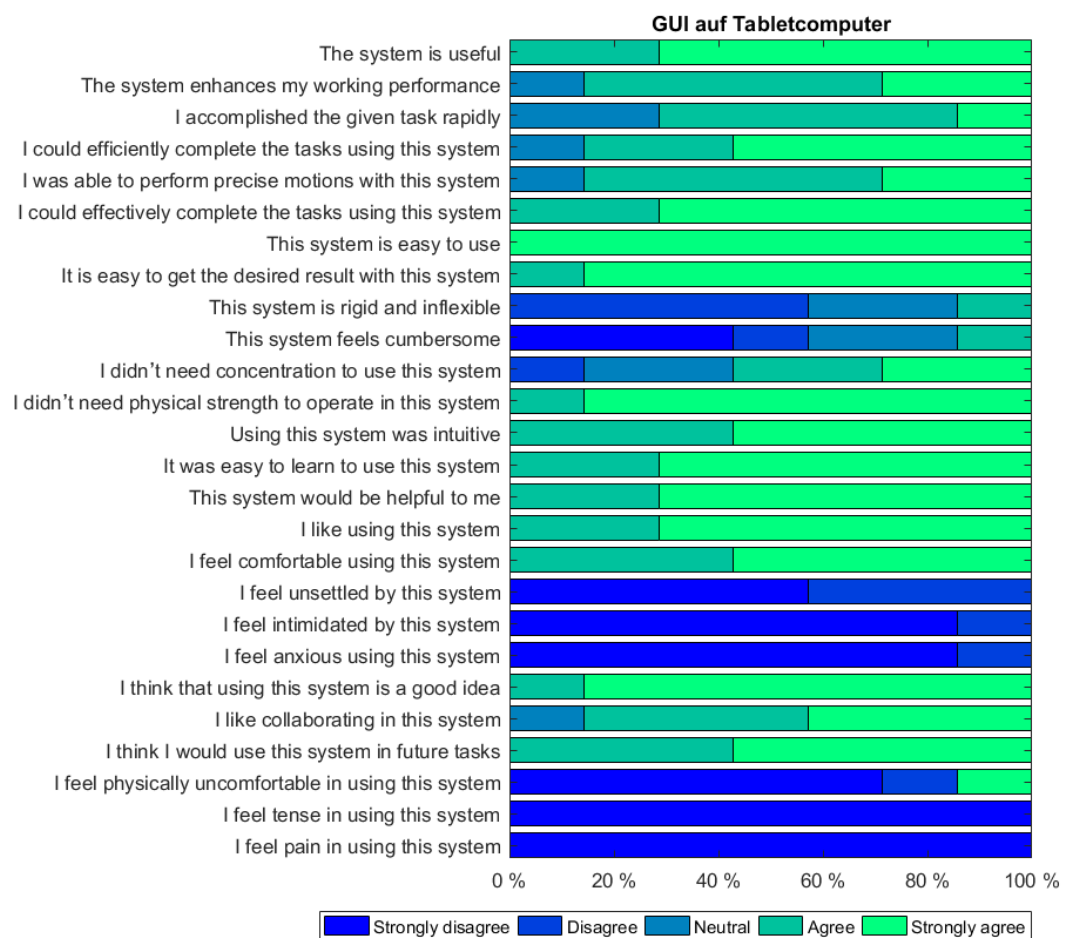


Abb. A.1.: Ergebnisse Fragebögen Gebrauchstauglichkeit GUI auf Tabletcomputer

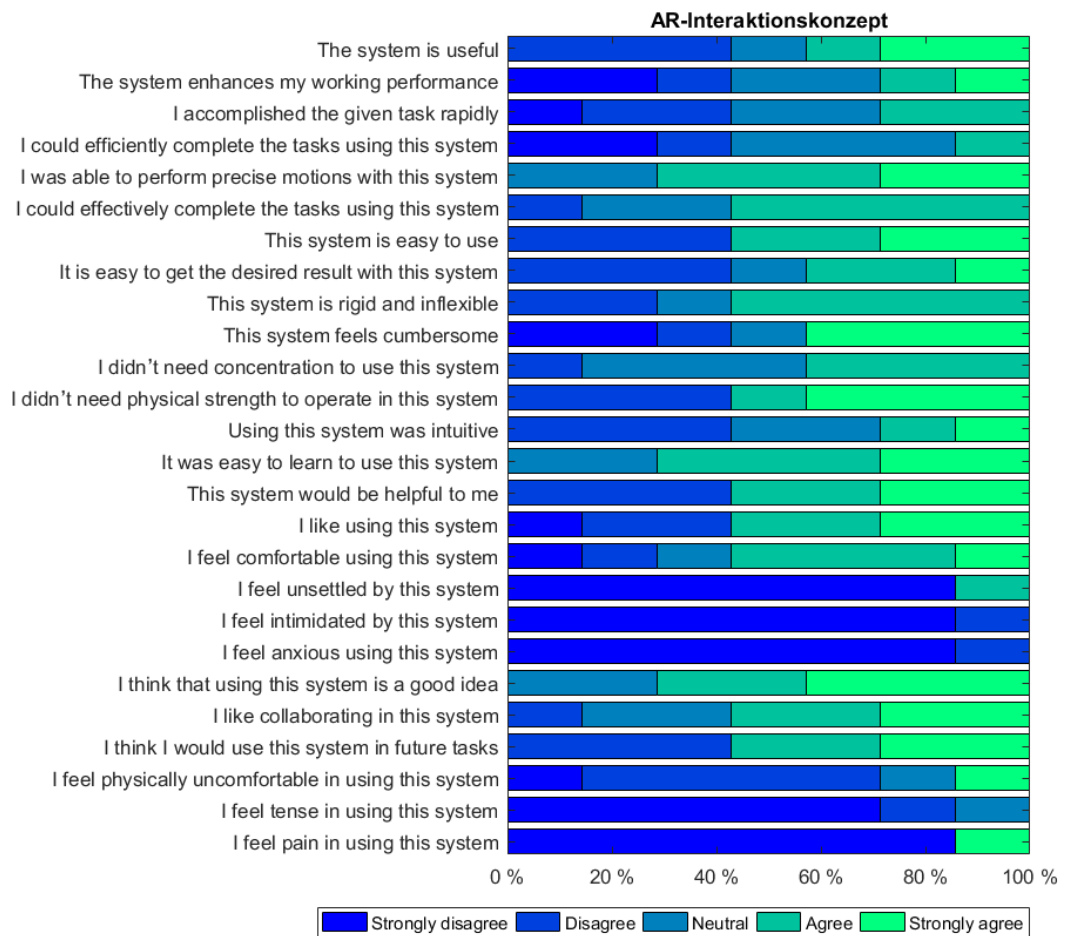


Abb. A.2.: Ergebnisse Fragebögen Gebrauchstauglichkeit AR-Interaktionskonzept

A.2 Fragebögen

Pre-study questionnaire

*Required

1. Please enter your identification code. *

The survey is anonymous, but to match answers from pre- and post-study questionnaires, some kind of identification is needed. Therefore, we ask you to enter a five-digit code. The first two digits of the code are the month you were born followed by the first three letters of your mother's first name. Example: If you were born on 9 March 1982 and your mother is called Anna, please enter 03ANN.

2. What year were you born? *

3. What is your gender? *

Mark only one oval.

- ☐ Female
- ☐ Male
- ☐ Other

4. What is the highest level of school you have completed or the highest degree you have received? *

Mark only one oval.

- ☐ No school-leaving qualification
- ☐ Secondary school certificate (Mittlere Reife)
- ☐ Highschool degree or equivalent (Abitur)
- ☐ Vocational training (Berufsausbildung)
- ☐ Bachelor degree or equivalent
- ☐ Master degree or equivalent (Diplom, Magister, ...)
- ☐ Doctoral degree or higher
- ☐ Other:

5. What is your handedness? *

Mark only one oval.

- ☐ Left
- ☐ Right
- ☐ No preference

6. Do you require vision aid of any kind (glasses or contact lenses)? *

Mark only one oval.

- ☐ Yes
- ☐ No

7. If yes, which kind?

Mark only one oval.

- ☐ Glasses
- ☐ Contact lenses
- ☐ Both
- ☐ Other: _____

8. How often have you used augmented/virtual reality devices in the past? *

Mark only one oval.

- ☐ Never
- ☐ Rarely
- ☐ Occasionally
- ☐ Regularly

9. How often have you used speech recognition in the past? *

Mark only one oval.

- ☐ Never
- ☐ Rarely
- ☐ Occasionally
- ☐ Regularly

10. How often have you used touch screens in the past? *

Mark only one oval.

- ☐ Never
- ☐ Rarely
- ☐ Occasionally
- ☐ Regularly

General

*Required

1. Please enter your identification code *

The first two digits of the code are the month you were born followed by the first three letters of your mother's first name. Example: If you were born on 9 March 1982 and your mother is called Anna, please enter 03ANN.

2. What system was used? *

Mark only one oval.

- ☐ Graphical User Interface (GUI) only
- ☐ GUI + Augmented Button
- ☐ GUI + Speech Recognition

Perceived usefulness

3. The system is useful

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

4. The system enhances my working performance

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

5. I accomplished the given task rapidly

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

6. I could efficiently complete the tasks using this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

7. I was able to perform precise motions with this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

8. I could effectively complete the tasks using this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

Perceived ease of use

9. This system is easy to use

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

10. It is easy to get the desired result with this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

11. This system is rigid and inflexible

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

12. This system feels cumbersome

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

13. I didn't need concentration to use this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

14. I didn't need physical strength to operate in this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

15. Using this system was intuitive

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

16. It was easy to learn to use this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

17. This system would be helpful to me

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

Emotions

18. I like using this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

19. I feel comfortable using this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

20. I feel unsettled by this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

21. I feel intimidated by this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

22. I feel anxious using this system

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

Attitude

23. I think that using this system is a good idea

Mark only one oval.

| | 1 | 2 | 3 | 4 | 5 | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

24. I like collaborating in this system

Mark only one oval.

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

25. I think I would use this system in future tasks

Mark only one oval.

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

Comfort

26. I feel physically uncomfortable in using this system

Mark only one oval.

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

27. I feel tense in using this system

Mark only one oval.

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

28. I feel pain in using this system

Mark only one oval.

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Strongly disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly agree |

Literatur

- [1] Batu Akan, Afshin Ameri, Baran Cürüklü und Lars Asplund. „Intuitive industrial robot programming through incremental multimodal language and augmented reality“. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, S. 3934–3939 (zitiert auf Seite 23).
- [2] Michael Bajura und Ulrich Neumann. „Dynamic registration correction in video-based augmented reality systems“. In: *IEEE Computer Graphics and Applications* 15.5 (1995), S. 52–60 (zitiert auf Seite 20).
- [3] Geoffrey Biggs und Bruce MacDonald. „A survey of robot programming systems“. In: *Proceedings of the Australasian conference on robotics and automation*, S. 1–3 (zitiert auf den Seiten 9, 10).
- [4] Oliver Bimber und Ramesh Raskar. *Spatial augmented reality: merging real and virtual worlds*. CRC press, 2005 (zitiert auf Seite 26).
- [5] John Brooke. „SUS-A quick and dirty usability scale“. In: *Usability evaluation in industry* 189.194 (1996), S. 4–7 (zitiert auf Seite 6).
- [6] Sebastian Brunner, Franz Steinmetz, Rico Belder und Andreas Dömel. „RAFCON: A graphical tool for engineering complex, robotic tasks“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, S. 3283–3290 (zitiert auf Seite 30).
- [7] John P. Chin, Virginia A. Diehl und Kent L. Norman. „Development of an instrument measuring user satisfaction of the human-computer interface“. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, S. 213–218 (zitiert auf Seite 6).
- [8] J. W. S. Chong, S. K. Ong, A. Y. C. Nee und K. Youcef-Youmi. „Robot programming using augmented reality: An interactive method for planning collision-free paths“. In: *Robotics and Computer-Integrated Manufacturing* 25.3 (2009), S. 689–701 (zitiert auf Seite 23).
- [9] Ute Claussen. *Programmieren mit OpenGL: 3D-Grafik und Bildverarbeitung*. Springer-Verlag, 2013 (zitiert auf Seite 22).

- [10]John J. Craig. *Introduction to robotics: mechanics and control*. 2004 (zitiert auf den Seiten 27, 28).
- [11]DIN EN ISO 9241-110:2008-09. *Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung*. 2008 (zitiert auf Seite 5).
- [12]DIN EN ISO 9241-11:1999-01. *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 11: Anforderungen an die Gebrauchstauglichkeit*. 1999 (zitiert auf den Seiten 5, 6).
- [13]Jill L. Drury, Jean Scholtz und Holly A. Yanco. „Awareness in human-robot interactions“. In: *IEEE International Conference on Systems, Man and Cybernetics*, 2003. Bd. 1, S. 912–918 (zitiert auf Seite 14).
- [14]Ralf Dörner, Wolfgang Broll, Paul Grimm und Bernhard Jung. *Virtual und augmented reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Springer-Verlag, 2014 (zitiert auf den Seiten 15, 17, 18, 21, 25, 26).
- [15]Mica R. Endsley. „Direct measurement of situation awareness: Validity and use of SAGAT“. In: *Situation awareness analysis and measurement 10* (2000) (zitiert auf Seite 14).
- [16]Mica R. Endsley. „The application of human factors to the development of expert systems for advanced cockpits“. In: *Proceedings of the Human Factors Society Annual Meeting*. Bd. 31. SAGE Publications, S. 1388–1392 (zitiert auf Seite 7).
- [17]Mica R. Endsley und Esin O. Kiris. „The out-of-the-loop performance problem and level of control in automation“. In: *Human factors 37.2* (1995), S. 381–394 (zitiert auf Seite 14).
- [18]Terrence Fong, Charles Thorpe und Charles Baur. „Collaboration, dialogue, human-robot interaction“. In: *Robotics Research*. Springer, 2003, S. 255–266 (zitiert auf Seite 10).
- [19]Subhashini Ganapathy, Glen J. Anderson und Delbert K. Marsh. „Techniques for mobile augmented reality applications“. In: (2016) (zitiert auf Seite 26).
- [20]Andre Gaschler, Maximilian Springer, Markus Rickert und Alois Knoll. „Intuitive robot tasks with augmented reality and virtual obstacles“. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. IEEE, S. 6026–6031 (zitiert auf Seite 23).
- [21]Claudio Gaz, Fabrizio Flacco und Alessandro De Luca. „Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach“. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. IEEE, S. 1386–1392 (zitiert auf Seite 34).

- [22]Michael A. Goodrich und Alan C. Schultz. „Human-robot interaction: a survey“. In: *Foundations and trends in human-computer interaction* 1.3 (2007), S. 203–275 (zitiert auf Seite 6).
- [24]Jens Grubert und Raphael Grasset. *Augmented Reality for Android application development*. Packt Publishing Ltd, 2013 (zitiert auf den Seiten 15, 20).
- [25]Matthias Haun. *Handbuch Robotik: Programmieren und Einsatz intelligenter Roboter*. Springer-Verlag, 2013 (zitiert auf Seite 9).
- [27]ISO/IEC 25012:2008. *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model*. 2008 (zitiert auf Seite 36).
- [28]David B. Kaber und Mica R. Endsley. „The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task“. In: *Theoretical Issues in Ergonomics Science* 5.2 (2004), S. 113–153 (zitiert auf Seite 14).
- [29]Jens Lambrecht und Jörg Krüger. „Spatial programming for industrial robots based on gestures and augmented reality“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012. IEEE, S. 466–472 (zitiert auf Seite 23).
- [30]Stanislao Lauria, Guido Bugmann, Theocharis Kyriacou und Ewan Klein. „Mobile robot programming using natural language“. In: *Robotics and Autonomous Systems* 38.3 (2002), S. 171–181 (zitiert auf Seite 12).
- [31]Joseph J. LaViola Jr. „A discussion of cybersickness in virtual environments“. In: *ACM SIGCHI Bulletin* 32.1 (2000), S. 47–56 (zitiert auf Seite 25).
- [32]James R. Lewis. „Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ“. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Bd. 36. SAGE Publications, S. 1259–1260 (zitiert auf Seite 6).
- [34]Paul Milgram, Shumin Zhai, David Drascic und Julius Grodski. „Applications of augmented reality for human-robot communication“. In: *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1993. Bd. 3. IEEE, S. 1467–1472 (zitiert auf Seite 8).
- [35]Paul Milgram, Haruo Takemura, Akira Utsumi und Fumio Kishino. „Augmented reality: A class of displays on the reality-virtuality continuum“. In: *Photonics for industrial applications*. International Society for Optics und Photonics, S. 282–292 (zitiert auf Seite 15).
- [36]Jakob Nielsen und Thomas K Landauer. „A mathematical model of the finding of usability problems“. In: *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, 1993. ACM, S. 206–213 (zitiert auf Seite 63).

- [37] Raja Parasuraman, Thomas B. Sheridan und Christopher D. Wickens. „A model for types and levels of human interaction with automation“. In: *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* 30.3 (2000), S. 286–297 (zitiert auf Seite 8).
- [38] Thomas Pettersen, John Pretlove, Charlotte Skourup, Torbjorn Engedal und T. Lkstad. „Augmented reality for programming industrial robots“. In: *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, S. 319 (zitiert auf Seite 23).
- [40] Helge Renkewitz, Mario Brandt und Thomas Alexander. „Benutzerzentrierte Entwicklung von Interaktionstechniken für die Erweiterte Realität“. In: *Ergonomie und Mensch-Maschine-Systeme*. Springer, 2008, S. 253–265 (zitiert auf Seite 14).
- [41] Cledja Rolim, Dieter Schmalstieg, Denis Kalkofen und Veronica Teichrieb. „Design Guidelines for Generating Augmented Reality Instructions“. In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2015*. IEEE, S. 120–123 (zitiert auf Seite 26).
- [42] Kenneth Salisbury, David Brock, Thomas Massie, Nick Swarup und Craig Zilles. „Haptic rendering: Programming touch interaction with virtual objects“. In: *Proceedings of the symposium on Interactive 3D graphics, 1995*. ACM, S. 123–130 (zitiert auf Seite 13).
- [43] Theodosios Sapounidis und Stavros Demetriadis. „Tangible versus graphical user interfaces for robot programming: exploring cross-age children’s preferences“. In: *Personal and ubiquitous computing* 17.8 (2013), S. 1775–1786 (zitiert auf Seite 11).
- [44] Jeff Sauro und James R. Lewis. *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann, 2016 (zitiert auf Seite 6).
- [45] Tamara von Sawitzky. „Entwicklung eines intuitiven Bedienkonzepts für Assistenzroboter unter Verwendung von Augmented Reality und Gestensteuerung“. Thesis. 2016 (zitiert auf Seite 51).
- [46] Raimund Schatz, Sebastian Egger und Kathrin Masuch. „The impact of test duration on user fatigue and reliability of subjective quality ratings“. In: *Journal of the Audio Engineering Society* 60.1/2 (2012), S. 63–73 (zitiert auf Seite 63).
- [47] J. Schmidtler, K. Bengler, F. Dimear und A. Campeau-Lecours. „A Questionnaire for the Evaluation of Physical Assistive Devices (QUEAD)“. In: *[submitted to] IEEE International Conference on Systems, Man, and Cybernetics 2017, Canada*. Series A Questionnaire for the Evaluation of Physical Assistive Devices (QUEAD). 2017 Published (zitiert auf Seite 54).
- [48] Jean Scholtz. *Evaluation methods for human-system performance of intelligent systems*. Report. National Institute of Standards und Technology, 2002 (zitiert auf den Seiten 2, 13).

- [49]Jean Scholtz. „Theory and evaluation of human robot interactions“. In: *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003. IEEE, 10 pp. (Zitiert auf den Seiten 6, 7).
- [50]Rolf Dieter Schraft und Christian Meyer. „The need for an intuitive teaching method for small and medium enterprises“. In: *VDI BERICHTE* 1956 (2006), S. 95 (zitiert auf Seite 1).
- [54]Franz Steinmetz und Roman Weitschat. „Skill parametrization approaches and skill architecture for human-robot interaction“. In: *IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, S. 280–285 (zitiert auf den Seiten 1, 29).
- [55]Franz Steinmetz, Annika Wollschläger und Roman Weitschat. „Human-robot interface with integrated framework for visual task-level programming and intuitive parametrization“. In: *[submitted to] IEEE International Conference on Robotics and Automation (ICRA)*, 2018 (zitiert auf Seite 30).
- [56]Hideaki Takeda, Nobuhide Kobayashi, Yoshiyuki Matsubara und Toyoaki Nishida. „Towards ubiquitous human-robot interaction“. In: *Working Notes for IJCAI-97 Workshop on Intelligent Multimodal Systems*, S. 1–8 (zitiert auf Seite 8).
- [57]Manfred Thüring und Sascha Mahlke. „Usability, aesthetics and emotions in human–technology interaction“. In: *International Journal of Psychology* 42.4 (2007), S. 253–264 (zitiert auf Seite 66).
- [58]Panagiota Tsarouchi, Athanasios Athanasatos, Sotiris Makris, Xenofon Chatzigeorgiou und George Chryssolouris. „High Level Robot Programming Using Body and Hand Gestures“. In: *Procedia CIRP* 55 (2016), S. 1–5 (zitiert auf Seite 12).
- [59]Richard Wetzel, Rod McCall, Anne-Kathrin Braun und Wolfgang Broll. „Guidelines for designing augmented reality games“. In: *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. ACM, S. 173–180 (zitiert auf Seite 26).
- [60]Matthias Wille, Philipp M. Scholl, Sascha Wischniewski und Kristof Van Laerhoven. „Comparing google glass with tablet-pc as guidance system for assembling tasks“. In: *11th International Conference on Wearable and Implantable Body Sensor Networks Workshops (BSN Workshops)*, 2014. IEEE, S. 38–41 (zitiert auf Seite 67).
- [61]Matthias Wille, Britta Grauel und Lars Adolph. „Strain caused by head mounted displays“. In: *Proceedings of the Human Factors and Ergonomics Society Europe* (2013), S. 267–277 (zitiert auf den Seiten 65, 66).
- [62]Michael F. Zaeh und Wolfgang Vogl. „Interactive laser-projection for programming industrial robots“. In: (Zitiert auf Seite 23).

Webpages

- [23]Google. *Google Glass Help - Tech Specs*. 2017. URL: <https://support.google.com/glass/answer/3064128?hl=en> (besucht am 1. Aug. 2017) (zitiert auf Seite 19).
- [26]Interaction Design Studio. *Research Process*. 2012. URL: <http://www.chloefan.com/design/fiddle/research.html> (besucht am 1. Aug. 2017) (zitiert auf Seite 16).
- [33]Microsoft Corporation. *HoloLens hardware details*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details (besucht am 1. Aug. 2017) (zitiert auf Seite 19).
- [39]PTC Incorporation. *VuMark*. 2016. URL: <https://library.vuforia.com/articles/Training/VuMark> (besucht am 1. Aug. 2017) (zitiert auf Seite 21).
- [51]Seiko Epson Corporation. *MOVERIO BT-200 Smart Glasses*. 2017. URL: <https://tech.moverio.epson.com/en/bt-200/> (besucht am 1. Aug. 2017) (zitiert auf den Seiten 19, 33).
- [52]Seiko Epson Corporation. *MOVERIO BT-300 Smart Glasses*. 2017. URL: <https://tech.moverio.epson.com/en/bt-300/> (besucht am 1. Aug. 2017) (zitiert auf Seite 19).
- [53]Sony Corporation. *SmartEyeglass*. 2017. URL: <https://developer.sony.com/devices/mobile-accessories/smarteyeglass/> (besucht am 1. Aug. 2017) (zitiert auf Seite 19).

Abbildungsverzeichnis

| | | |
|------|--|----|
| 2.1 | Mixed Reality Kontinuum nach [35] (aus [14]) | 15 |
| 2.2 | Erzeugung von AR-Effekten durch Projektion beim Billard [26] | 16 |
| 2.3 | Stereosehen mit einem Stereodisplay [14] | 17 |
| 2.4 | Schematischer Aufbau eines OSTs (aus [14]) | 18 |
| 2.5 | Beispiel für einen Marker zum kamerabasierten Tracking [39] | 21 |
| 2.6 | Renderingpipeline in OpenGL [9] | 22 |
| 2.7 | Beispiel eines 3DOF-Roboters [10] | 28 |
| 3.1 | Überblick über die Komponenten von RAZER [55] | 30 |
| 3.2 | GUI bei der Erstellung eines neuen Skills | 31 |
| 3.3 | Sequenz aus drei parametrisierten Skills | 32 |
| 3.4 | Epson Moverio BT-200 Smart Glasses [51] | 33 |
| 3.5 | Gelenkkonfiguration des KUKA LWR | 34 |
| 3.6 | Basisorientiertes und eingabeorientiertes Eingabekonzept | 37 |
| 3.7 | Entwürfe der grafischen Anzeige für das Interaktionskonzept . . | 38 |
| 3.8 | Komponenten des AR-Interaktionskonzepts | 39 |
| 3.9 | Dialoghierarchie RAZER | 42 |
| 3.10 | VuMark mit Code-Elementen | 47 |
| 3.11 | Darstellung der Koordinatensysteme am Mensch-Roboter-System | 49 |
| 3.12 | Anzeige der Interaktionsoptionen am Roboter | 51 |
| 3.13 | Anzeige einer Fehlermeldung am Roboter mit Sichtfeld der Datenbrille | 52 |
| 3.14 | Sequenzdiagramm der Initialisierung | 53 |
| 3.15 | Sequenzdiagramm bei Eingabe | 54 |
| 3.16 | Versuchsablauf der Nutzerstudie | 58 |

| | | |
|-----|---|----|
| 4.1 | Ergebnisse der Fragebögen zur Gebrauchstauglichkeit | 60 |
| 4.2 | Bearbeitungszeiten | 61 |
| A.1 | Ergebnisse Fragebögen Gebrauchstauglichkeit GUI auf Tablet- computer | 72 |
| A.2 | Ergebnisse Fragebögen Gebrauchstauglichkeit AR-Interaktions- konzept | 73 |

Tabellenverzeichnis

| | | |
|-----|---|----|
| 2.1 | Vergleich von OST-Brillen | 19 |
| 3.1 | Vergleich der AR-Frameworks ARToolKit und Vuforia | 46 |
| 4.1 | Ergebnisse der Fragebögen zur Gebrauchstauglichkeit | 59 |
| 4.2 | Bearbeitungszeiten | 61 |

Abkürzungsverzeichnis

ABService Augmented-Button-Service

AOA Android Open Accessory

API Application Programming Interface

AR Augmented Reality

DLR Deutsches Zentrum für Luft- und Raumfahrt

DLR Deutschen Zentrum für Luft- und Raumfahrt

DOF Degrees of Freedom

GUI Graphical User Interface

IDE Integrated Development Environment

IMU Inertial Measurement Unit

IOService Input-Output-Service

JSON JavaScript Object Notation

KMU kleine und mittlere Unternehmen

MR Mixed Reality

MRI Mensch-Roboter Interaktion

NUI Natural User Interface

OST Optical-See-Through

PdV Programmieren durch Vormachen

REST Representational State Transfer

SDK Software Development Kit

TUM Technische Universität München

TUI Tangible User Interface

VR Virtual Reality

VST Video-See-Through